AD-A069 744 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 15/7
A COMPUTERIZED TECHNIQUE FOR SCHEDULING MILITARY AIRLIFT COMMAN--ETC(U)
MAR 79 G P MILNE, R K COFFEY

UNCLASSIFIED AFIT/GST/SM/79M-3 NI

1 OF 2
AD
A069744

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

LEVEL II

AD A069744

DDC FILE COPY

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFIT/GST/SM/79M-3	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): A COMPUTERIZED TECHNIQUE FOR SCHEDULING MILITARY AIRLIFT COMMAND CT-39 OPERATIONAL SUPPORT AIRLIFT MISSION	5. TYPE OF REPORT & PERIOD COVERED: M.S. Thesis
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Roger K. Coffey, Major, USAF; George P. Milne, Major, USAF	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE: March 1979
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

DDC RECEIVED JUN 7 1979 C

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

Joseph P. Hipps, Major, USAF 23 MAR 1979
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Airlift
Scheduling
Modeling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research was conducted to develop a computerized technique which demonstrates a methodology that will enable the Military Airlift Command Administrative Airlift Division (MAC/DOOF) to quickly prepare good CT-39 operational support airlift mission initial schedules.

The essential elements of the present manual scheduling process and the characteristics of a good schedule were identified.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

next page

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1

A heuristic scheduling algorithm was developed and incorporated into a computerized scheduling model. Schedules prepared using the model were compared to schedules produced manually by MAC/DOOF.

The results of the comparison showed that the model was able to assist the researchers in producing schedules which were more effective than those produced by MAC/DOOF. However, the comparison did not consider the different environments in which the schedules were prepared. When the environmental factors were considered, it became clear that the model would be of most value to MAC/DOOF if it would allow them to wait longer before beginning preparation of each daily schedule.

A benefit-cost analysis of the model was not performed; however, the recommendations include items that will assist MAC/DOOF in performing such an analysis.

Accession For	
NTIS GRA&I	✓
DDC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

A COMPUTERIZED TECHNIQUE FOR SCHEDULING
MILITARY AIRLIFT COMMAND CT-39
OPERATIONAL SUPPORT AIRLIFT MISSIONS

THESIS

George P. Milne Major USAF
Roger K. Coffey Major USAF

AFIT/GST/SM/79M-3

Approved for public release; distribution unlimited.

79 05 30 203

14 AFIT/GST/SM/79M-3

6 A COMPUTERIZED TECHNIQUE FOR SCHEDULING MILITARY AIRLIFT COMMAND CT-39 OPERATIONAL SUPPORT AIRLIFT MISSIONS.

9 Master's ~~T~~HESIS,

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

12 179 p.

by

10 George P. Milne   Roger K. Coffey
~~Major   USAF   Major   USAF~~

Graduate Strategic and Tactical Sciences

11 Mar~~ch 19~~79

Approved for public release; distribution unlimited.

012 225

slt

Preface

This research was performed to develop the methodology for automating the initial portion of the Military Airlift Command Administrative Airlift Division (MAC/DOOF) CT-39 operational support airlift mission scheduling process. The computerized scheduling model we have developed demonstrates that this automation is possible, but we have not conducted a benefit-cost analysis to determine if it is desirable.

In order to allow MAC/DOOF to conduct such an analysis, we have attempted to include thorough documentation of the model. The report contains a program listing, a user's guide, descriptions of each routine, and a number of logic flow diagrams. The program is written in SIMSCRIPT II.5, which is a somewhat self-documenting language, and comment cards are included throughout the program.

We wish to express our appreciation to Major Thomas Griesser for serving as our primary contact at Headquarters MAC, and to Captain Patrick Terry for providing us with the schedule and travel request data needed to verify and validate our model. We also want to thank Major Will Owings and his staff for answering our many questions about MAC/DOOF Planning Branch scheduling procedures. Last, but not least, we want to thank Dr. Edward J. Dunne for allowing us so much freedom in our approach to this research.

George P. Milne and Roger K. Coffey

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This research was conducted to develop a computerized technique which demonstrates a methodology that will enable the Military Airlift Command Administrative Airlift Division (MAC/DOOF) to quickly prepare good CT-39 operational support airlift mission initial schedules.

The essential elements of the present manual scheduling process and the characteristics of a good schedule were identified. A heuristic scheduling algorithm was developed and incorporated into a computerized scheduling model. Schedules prepared using the model were compared to schedules produced manually by MAC/DOOF.

The results of the comparison showed that the model was able to assist the researchers in producing schedules which were more effective than those produced by MAC/DOOF. However, the comparison did not consider the different environments in which the schedules were prepared. When the environmental factors were considered, it became clear that the model would be of most value to MAC/DOOF if it would allow them to wait longer before beginning preparation of each daily schedule.

A benefit-cost analysis of the model was not performed; however, the recommendations include items that will assist MAC/DOOF in performing such an analysis.

A COMPUTERIZED TECHNIQUE FOR SCHEDULING MILITARY AIRLIFT COMMAND CT-39 OPERATIONAL SUPPORT AIRLIFT MISSIONS

I. INTRODUCTION

Background

Operational support aircraft perform Air Force-directed missions that include the priority movement of people and cargo with time, place, or mission-sensitive requirements. Air Force Regulation (AFR) 60-23 assigns to the Military Airlift Command (MAC) the responsibility for scheduling and managing all operational support airlift missions in the continental United States (CONUS) (Ref 1:1).

Air Force personnel submit their requests for CONUS operational support airlift to MAC through their unit mission request validators. Other eligible Department of Defense (DOD) personnel submit their requests to MAC through the office of the USAF Vice Chief of Staff (Ref 1:1).

To support the requests of personnel traveling in groups of six or less, MAC operates a fleet of 100 CT-39 aircraft from 15 CONUS Air Force bases. The MAC Administrative Airlift Division (MAC/DOOF) schedules and manages this fleet. CT-39 scheduling involves the MAC/DOOF Planning Branch, the Requirements Branch, and the Change Section.

MAC/DOOF personnel manually prepare daily CT-39 mission schedules. The Planning Branch must prepare an initial schedule at least three days in advance. The Planning

Branch sends this initial schedule to the Requirements Branch, which notifies travelers of support or nonsupport for their requests at least two days prior to the day of travel. This early notification allows travelers whose requests cannot be supported to obtain other means of transportation (Ref 1:6). The Requirements Branch then sends the schedule to the Change Section, which makes minor adjustments based on late cancellations and on last-minute receipt of top-priority requests. The Change Section publishes the final schedule at least one day prior to the day of travel so that the CT-39 operating locations will have sufficient time to notify the aircrews selected to fly the missions (Ref 1:6).

Problem

The Planning Branch scheduler must prepare seven initial schedules during each work week. Preparation of each initial schedule normally spans across two work days. For an average initial schedule, approximately 50 aircraft must be allocated among over 300 requests that represent more than 500 travelers (Ref 9:1-1, 1-2). This demand exceeds the available airlift capacity. As a result, the scheduler must make numerous value judgments in determining which is the best combination of requests that can be supported.

The current manual method of scheduling is time-consuming and does not allow for rapid evaluation of alternative scheduling strategies. We propose that automating the CT-39 initial schedule preparation process, or at least

part of it, will help MAC/DOOF make better use of their resources by:

a. Reducing the time required to produce an initial schedule.

b. Providing a means for quick comparison of alternative scheduling strategies.

If a computerized scheduling model can be developed, the time required to produce a schedule will be reduced. The issue then becomes the quality of the schedule produced by the model. The relevant questions are:

a. Can a computerized scheduling model be developed?

b. Does the model produce a good schedule?

Objectives

Primary Objective. The primary objective is to develop a computerized model that demonstrates a methodology which will enable MAC/DOOF personnel to quickly prepare a good CT-39 operational support airlift mission initial schedule.

Secondary Objectives. We consider these to be the major steps necessary to meet our primary objective:

a. Identify the essential elements of the CT-39 initial scheduling process.

b. Identify the characteristics of a good schedule.

c. Develop a scheduling model.

d. Test the performance of the model.

Scope and Limitations

The model is designed specifically to assist a scheduler in preparing a CT-39 initial schedule. Although it may be adapted to assist in making schedule revisions, it is not designed to perform that function.

The model is programmed in SIMSCRIPT II.5, which is not compatible with current MAC/DOOF computer software. However, there is a SIMSCRIPT II.5 compiler at MAC headquarters which may be made available to MAC/DOOF.

As will be developed, no analytical solution exists to a scheduling problem of this complexity. As a result, there is no optimal standard against which to test the performance of the model. Schedules produced using the model have been tested against actual schedules produced by the MAC/DOOF Planning Branch.

Overview

Chapter II addresses the essential elements of the CT-39 scheduling process and the characteristics of a good schedule. Chapter III discusses the computerized model which has been developed for scheduling CT-39 operational support airlift missions, and it examines each major routine in the program.

In Chapter IV, we describe how the model can be used to create an initial schedule. We then evaluate the

validity of the model by comparing model output to schedules developed by the MAC/DOOF Planning Branch.

Chapter V presents our conclusions and recommendations.

II. THE CT-39 OPERATIONAL SUPPORT AIRLIFT INITIAL SCHEDULE PREPARATION PROCESS

This chapter defines the boundaries, restrictions, and measures of effectiveness for the process used by the MAC/DOOF Planning Branch to prepare a CT-39 initial schedule. The CT-39 initial schedule is a plan that directs an itinerary for each aircraft. This itinerary is a series of legs between airfields. Each leg has a departure time and an arrival time. On each leg, the aircraft may possibly support passengers who have submitted travel requests.

Preparation of this schedule may be viewed as a resource allocation process with three essential elements. The resources are the aircraft, the demand for resources comes from travel requests, and the allocation of resources is performed by the MAC/DOOF Planning Branch schedulers. This chapter examines each of these essential elements.

Aircraft

Aircraft are the resources that the scheduler must allocate over time to satisfy travel requests. This section describes these resources and the important constraints on their use.

The CT-39 Sabreliner is a small jet transport aircraft. MAC Sabreliners are flown by a crew of two pilots. Most CT-39s can accommodate five passengers, but a few can carry six. Planned modifications will standardize the passenger

capacity at five (Ref 6).

The Sabreliner has maximum airspeed of over 500 knots; however, normal cruise airspeed is approximately 410 knots. Maximum range is approximately 1700 nautical miles (NM) (Ref 12:124).

MAC has a fleet of 100 Sabreliners assigned to 15 detachments at the operating locations shown in Table I.

Table I

Operating Locations and Numbers of MAC CT-39s

Operating Location	Number of Aircraft
Andrews AFB, MD	10
Barksdale AFB, LA	4
Bergstrom AFB, TX	4
Eglin AFB, FL	5
Kirtland AFB, NM	5
Langley AFB, VA	13
Maxwell AFB, AL	4
McClellan AFB, CA	5
Norton AFB, CA	6
Offutt AFB, NB	12
Peterson AFB, CO	5
Randolph AFB, TX	8
Scott AFB, IL	6
Shaw AFB, SC	4
Wright-Patterson AFB, OH	9

(Ref 6)

Only about half of this fleet is available for operational support airlift each day. The other aircraft may be scheduled for alert duty, training missions, or maintenance (Ref 6).

Aircraft are constrained by their unrefueled ranges. As a safety-of-flight measure, MAC/DOOF adheres to the following policy concerning refueling:

a. The maximum planned non-stop flying time between refuelings is 3 hours and 15 minutes.

b. The maximum planned flying time for a mission segment that includes an enroute stop with no refueling is 2 hours and 30 minutes (Ref 5).

A minimum of one hour ground time is scheduled when refueling is required; otherwise, a minimum of 30 minutes ground time is scheduled (Ref 10).

Flight routes between airfields are normally by the most direct routing compatible with Federal Aviation Administration (FAA) air traffic control procedures. Planned flying times are based on estimates of the times required for climb, cruise, descent, approach, and landing. Cruise time estimates are based on the spherical point-to-point distances between airfields, a cruise airspeed of 410 knots, and approximate seasonal winds.

From 1 May to 30 September, MAC/DOOF estimates the winds aloft over the entire CONUS to be directly from the west at 25 knots. At all other times, the winds are estimated to be from the west at 65 knots (Ref 10).

The utilization of each Sabreliner is limited by the maximum allowable duty day of the aircrew which must fly it. Normally, only one aircrew is scheduled to fly any particular aircraft on a given day. Crew duty begins when the pilots report to begin preparation for the mission. This is normally two hours prior to their first scheduled takeoff. It ends when the crew parks the aircraft after the last landing of the mission. The maximum duty day is normally 12 hours; however, if crew duty starts between 0600 and 1000 local time, the maximum crew duty day is 14 hours. Regardless of crew duty start time, the maximum duty day is 16 hours for a mission dedicated to the USAF Chief of Staff or Vice Chief of Staff, or to a four-star general commander of a USAF major air command (Ref 8:2-1).

Aircrews impose an additional constraint on aircraft which remain overnight (RON) away from their bases of assignment. Crew rest requirements, ground transportation requirements, and mission planning duties result in a normal ground time of 15 hours at an RON base. Hence the aircraft may not be available to support requests for travel early on the next schedule day.

Aircraft are constrained to operate from airfields. For the MAC/DOOF scheduler, the most significant characteristics of an airfield are:

a. Identification.

b. Geographical location.

c. Difference between local time and Greenwich

Mean Time (GMT).

d. Runway length.

e. Aircraft services, hours of operation, and other elements of the operational environment.

In addition to its name, a major airdrome may be referred to by a four-letter identifier assigned by the International Civil Aviation Organization (ICAO). A CONUS airdrome with no ICAO identifier is assigned a three-character alphanumeric identifier by the FAA.

Geographical locations establish the point-to-point distance and hence the planned flying time between airfields. Planned flying time determines whether or not refueling will be required on a mission between two particular airfields.

Since times on the MAC/DOOF CT-39 schedule are in GMT, knowledge of the difference between local time and GMT for the base at which the mission originates is required to establish crew duty start time and maximum crew duty day.

MAC has specified the following minimum runway lengths for CT-39 operations:

a. 6000 feet for dry conditions.

b. 7000 feet for wet conditions (Ref 8:3-2).

Schedulers obtain airfield status information such as runway lengths, aircraft services available, and hours of operation from DOD Flight Information Publications, Notices to Airmen, and Special Notices.

Travel Requests

Travel requests are the demand for use of CT-39 resources. DOD specifies who can travel aboard operational support aircraft. Eligible USAF personnel submit travel requests to MAC through their unit mission request validators. Eligible personnel from agencies outside the Air Force must submit their requests to MAC through the office of the USAF Vice Chief of Staff (Ref 1:2).

When demand for operational support airlift exceeds the capability, resources must be allocated by priority. Each travel request is assigned a priority based on a system established by HQ USAF (Ref 1:3). This priority system is detailed in Table II. Each travel request contains the following information:

a. Origin ICAO identifier.

b. Destination ICAO identifier.

c. Earliest acceptable Julian date and GMT for departure.

d. Latest acceptable Julian date and GMT for arrival.

e. The number of people traveling together.

f. The priority, rank, branch of service, and name of the requester.

g. The name, home phone number, and duty phone number of the mission request validator or other person whom MAC can contact regarding the request.

Table II

Operational Support Airlift Priority System

Priority	Travel authority and purpose
1	Directed by HQ USAF as a flight of emergency nature or vital to the national interest.
2	Directed by HQ USAF/CV to conduct extremely urgent official business.
3	To transport general officers and civilians of comparable grade conducting urgent official business.
4	Directed by HQ USAF (DCS or equivalent levels) and command sections of MAJCOMs or SOAs as a flight required to conduct urgent official business.
5	Directed by HQ USAF/IG or AFISC to transport personnel conducting an IG inspection.
6	Directed by MAJCOM IG to transport personnel conducting an IG inspection.
7	Directed by a MAJCOM or SOA to transport personnel conducting a standardization evaluation.
8	Directed by HQ USAF (DCS or equivalent level), MAJCOM, or SOA as a flight required to conduct essential official business.
9	Directed by a numbered air force, AFR region, ALC, TAG, TTC, and MTCs as a flight required to conduct essential official business.
10	Directed by an Air Division or center (non-SOA) as a flight required to conduct essential official business.
11	Directed by a wing as a flight required to conduct essential official business.
12	All other requests to conduct routine official business.

(Ref 1:3)

The deadline for receipt of priority 4-12 requests is three days (two of which must be duty days) prior to the day of travel. The deadline for priority 3 requests is two days (one of which must be a duty day) prior to the day of travel (Ref 1:6).

During the second quarter of calendar year 1978, MAC received 29,512 requests to transport 51,336 personnel by CT-39. Table III shows the number of requests of each priority received and the percent supported.

Table III

CT-39 Requests Submitted and Percent Supported

(April-June 1978)

Priority	Requests Submitted	Percent Supported
1	50	100
2	76	99
3	4093	98
4	2298	38
5	528	20
6	345	29
7	475	35
8	5250	30
9	6150	20
10	640	24
11	502	43
12	9096	28

(Ref 9:1-2)

MAC/DOOF Schedulers

The MAC/DOOF schedulers must decide which is the best combination of travel requests that can be supported with the Sabreliners available on a given day. AFR 60-23 specifies the following scheduling procedures:

> Priority 1 or 2 requests will be supported within resource capability, regardless of time of submission.
>
> MAC will schedule support for travel requests on a cost effective basis as necessary to meet mission requirements, and using the priority system outlined here. (Note: See Table II.)
>
> (1) To make the best use of the aircraft MAC will consider these options: moving individual travelers on a scheduled team travel mission; adjusting routes; or recommending other travel times or dates. To make best use of resources, MAC will operate airlift missions primarily on a "pick-up" and "drop-off" basis.
>
> (2) If these efforts do not meet the mission requirements, MAC will schedule the missions according to the priority system; then by DV code, rank, and date of rank, as specified in the Flight Information Publication Planning Document and in AFR 102-8 (Ref 1:2).

AFR 60-23 also directs that, unless it is essential to meet mission requirements, MAC will not:

a. Dedicate an airlift mission to a single user, unless this is a more efficient use of resources.

b. Schedule a mission that requires an aircraft to remain overnight away from its home station, unless no other arrangement will meet the user's travel requirement (Ref 1:3).

These procedures and restrictions allow some latitude for scheduler judgment. MAC/DOOF personnel consider a good

initial schedule to have the following properties:

a. Observes all scheduling procedures, restrictions, and policies.

b. Supports all priority 1-3 travel requests.

c. Carries as many other passengers as possible (Ref 2).

This measure of effectiveness for an initial schedule is based on the fact that nearly every CT-39 mission supports at least one priority 1-3 request. Most lower priority requests are satisfied because they are compatible with the routings and times of travel dictated by the priority 1-3 requests.

To begin his manual scheduling process, the scheduler normally makes these preparations:

a. Sorts the requests. Each request is on a separate piece of paper. The priority 1-3 requests are placed in one stack, and the priority 4-12 requests are separated by departure base time zones into four stacks. Within each time zone stack, the requests are ordered by departure point and then by departure time.

b. Checks the requests.

1. Some requests have typographical errors in the origin and destination identifiers. The mission request validator or other contact must be called to clarify the requested route of travel.

2. Some priority 1-3 requests have only one minute difference between the earliest acceptable departure

time and the latest acceptable arrival time. This is a convention by which high priority travelers indicate an inflexible arrival or departure time. The scheduler interprets whichever time on the request ends in zero or five as the inflexible time and adjusts the other time by the estimated flying time.

c. Confirms the number, locations, and home stations of available aircraft and determines if any of the available aircraft have six seats.

After completing these preparations, the scheduler reviews all the priority 1-3 requests and formulates a general plan to meet these requests. A major consideration in this plan is the requirement for each aircraft to return to its home station within the aircrew's duty day limitations. Some requests require too much flying to be supported by a Sabreliner which originates and terminates its itinerary at the same airfield. The scheduler's options to satisfy such requests are:

a. Use an aircraft that has remained overnight away from its home station.

b. Schedule an aircraft to remain overnight away from its home station.

c. Interplane the request.

Interplane is a MAC/DOOF term for the process of using more than one airplane to support a request. An example is for a request for travel from Andrews AFB to Norton AFB to be supported by:

a. One CT-39 flying the requester's party from Andrews AFB to Offutt AFB.

b. Another CT-39 carrying the requester's party from there to Norton AFB (Ref 10).

Interplane is not the most preferred method of supporting travel requests because unforeseen maintenance or weather difficulties that affect one Sabreliner can negate the productivity of two aircraft. As a result, interplane is not considered as a method to support priority 4-12 requests.

Another major consideration in developing this general plan is to combine as many requests as possible. The scheduler asks the Requirements Branch to investigate with travel requesters possible changes to their arrival and departure times in an effort to allow more requests to be satisfied or more passengers to be transported.

After determining how to support as many priority 1-3 requests as possible, the scheduler searches the priority 4-12 requests and decides how to make the most productive use of any empty seats or any unused crew duty time for each available aircraft.

The scheduler is now ready to expand the general plan into a specific set of itineraries for the available Sabreliners. Late cancellations and receipt of additional requests may require scheduling without an opportunity to revise the general plan. Appendix A illustrates how the deadline for request submission overlaps the schedule preparation process for a normal day. The scheduler ultimately

arrives at what is considered to be the best possible combination of itineraries; however, there may have been no opportunity to test this schedule against other possible combinations of itineraries.

When the scheduler defines the itinerary for each available Sabreliner and identifies the passengers to be carried on each leg of that itinerary, the initial schedule is completed. The Planning Branch sends the schedule to the Requirements Branch, which notifies travelers of support or nonsupport for their requests. The schedule may still be revised, but MAC must make every effort to avoid disrupting a confirmed travel schedule once the requester has been notified of support (Ref 1:3).

Summary

This chapter has identified the essential elements of the CT-39 initial schedule preparation process as aircraft, travel requests, and schedulers. It has also identified the characteristics of a good schedule. The following chapter describes how this information has been incorporated into a computerized scheduling model.

III. THE MODEL

This chapter presents a basic description of the model developed to produce a CT-39 operational support airlift mission initial schedule. A complete program listing is contained in Appendix D, and a user's guide is contained in Appendix B.

The purpose of the model is to produce a schedule that:

a. Observes all scheduling procedures, restrictions, and policies.

b. Supports as many priority 1-3 requests as possible.

c. Supports as many priority 4-12 passengers as possible.

Components of the actual system included in the model are airfields, aircraft, travel requests, and the MAC/DOOF Planning Branch schedulers. Airfields, aircraft, and travel requests are represented as entities. The schedulers are modeled by a set of heuristic algorithms.

A critical element in the model is the algorithm used to represent the scheduler's aircraft itinerary development process. Several models use algorithms similar to the one we have employed, and a discussion of these models follows.

Mobility and Airlift Models

The algorithm we have developed for scheduling limited CT-39 resources to support competing travel requests has

elements similar to algorithms used in models of intertheater strategic mobility systems and intratheater tactical airlift systems.

Most strategic mobility models are concerned with the deployment of general purpose forces from the CONUS to the theater of conflict. The earliest of these models specified a desired level of effectiveness and used linear programming techniques to determine a least-cost mix of airlift and sealift forces to meet the constraints and assumptions input by the user. The models also developed a set of deployment plans for each contingency area; however, these plans only specified the tonnage to be moved by each type of transport vehicle and the route over which it was to be moved. Considerable additional planning was required to develop an actual movement schedule from these deployment plans (Ref 4:III-3 through III-5).

More advanced linear programming models still suffered from some basic limitations, Among these were:

a. The problem of defining an optimal solution. Most requirements could not be adequately captured by a single objective function.

b. The problem of non-integer solutions. While fractions of ships and airplanes were not meaningful, the problems were too complex for integer programming solution (Ref 4:III-8).

The linear programming distribution models (such as the transportation and transshipment models), which always yield

integer solutions when feasible solutions exist, were not powerful enough because their necessary assumptions were too stringent for the actual problem. The two most visible assumptions violated by the actual problem were the requirements that:

a. The commodities being shipped either must be the same or must be substitutes for one another.

b. The cost of transporting units of a commodity from a particular source to a particular destination must be directly proportional to the number of units shipped (Ref 3:112-113).

It is apparent that the distribution models are also not powerful enough for the operational support airlift mission scheduling problem. The commodities being shipped in this instance are people of various capabilities, ranks, and travel priorities; and they are seldom substitutes for one another. Even if all travelers were substitutes for one another, the cost of transporting one of them directly between two points would be at best only slightly less than the cost of transporting five of them directly between the same two points.

Even the most advanced linear programming strategic mobility models did not track individual aircraft. Aircraft were aggregated and treated in terms of productivity per day (Ref 4:III-21). The advent of more powerful computers led to simulation models which allowed analysts to trade the optimality feature of linear programming models for the

detailed movement information that could be extracted from deployment simulations. Analysts had concluded that applica- of heuristic decision rules to a deployment simulation could yield near-optimal solutions; so little optimality was sur- rendered in the exchange for much more useful scheduling information (Ref 4:III-17).

An early simulation model was QTYP, which used a simple heuristic rule to schedule force movements. The forces were moved in a priority order, and the priorities were based on a required delivery date (Ref 4:III-15).

The Interactive Strategic Deployment Model (ISDM) is a recent logistics simulation model that also uses a set of heuristic decision rules to assign transportation resources to move cargo (Ref 4:III-19). Although ISDM tracks indivi- dual ships, aircraft are still aggregated as in the linear programming models. Cargo considered in this model has attri- butes of priority, earliest date of availability, and required delivery date. Once again, required delivery date is the fac- tor that determines the sequencing of forces (Ref 4:III-23).

In his description of ISDM, Hoeber addresses the pro- blem of selecting scheduling rules for cargo with inconsis- tent availabilities and priorities (that is, lower-priority cargo is available for shipment before the top-priority cargo is available to move). He illustrates how a schedul- ing rule that moves the highest priority cargo available when the transportation resources are ready may make better use of resources and time than a rule which schedules

exclusively in priority order (Ref 4:III-24, 25).

Sherman developed a model for scheduling tactical airlift missions. He recognized that optimizing the schedule for an entire day's missions necessitated the solution of an extremely large number of combinatorial problems. He elected to design an algorithm that used dynamic programming techniques to schedule one mission at a time (Ref 11:12). Although this was a suboptimization approach, Sherman concluded that schedules produced by his automated algorithm were significantly better than schedules produced manually in tests using the same data (Ref 11:31).

We found the algorithms used in these earlier models to be similar to the one we have developed which:

a. Assigns itineraries to Sabreliners one at a time.

b. Ranks requests according to a modified priority system.

This system orders requests by the traveler's priority, but within each priority group the requests are ordered by NLT time rather than by DV code, rank, and date of rank.

The details of the algorithm will be supplied in the descriptions of the model routines that follow. A list of all model routines is presented in Table IV. Simplified logic flow diagrams are included for most routines.

Table IV

Model Routines

1.	Preamble
2.	Main
3.	Read Data
4.	T-39 Scheduler
5.	Leg Data
6.	Refuel
7.	Position
8.	Interplane
9.	Print Schedule

Preamble

The Preamble specifies the attributes of the physical elements of the system and the relationships among those elements. Attributes of the entities defined by the Preamble are listed in Table V. Relationships among these entities are discussed below.

Airfields. Two sets of airfields are created by the model. The first set is called the Base File. It contains all the airfields for which the user has supplied information to the model. These airfields are called Bases, and they are created by the Read Data routine. Each Base owns a set containing all of the aircraft located there at the beginning of the schedule day. Bases are ranked alphabetically in the Base File.

Table V

Attributes of Entities Defined by Preamble

1. Base
   a. Name
   b. Identifier (ICAO or FAA)
   c. North latitude
   d. West longitude
   e. Standard time difference from GMT
   f. Daylight saving time difference from GMT
   g. Local time correction (either e. or f. above)

2. RF Base
   a. Identifier (ICAO or FAA)
   b. Flight time and fuel required from origin
   c. Flight time and fuel required to destination
   d. Total time (b. + c. + 1 hour of ground time)
   e. Passenger value

3. Sabreliner
   a. Home station
   b. Crew duty start time
   c. Maximum crew duty day
   d. Duty day
   e. Seats available

4. Leg
   a. Origin
   b. Destination
   c. Departure time
   d. Enroute time
   e. Arrival time
   f. Fuel required

5. Travel request
   a. Origin
   b. Destination
   c. Not-earlier-than date for departure (NET date)
   d. NET time
   e. Not-later-than date for arrival (NLT date)
   f. NLT time
   g. Passenger load
   h. Passenger priority
   i. Passenger Distinguished Visitor code (DV code)
   j. Passenger name
   k. Passenger rank

The second set of airfields is called the RF Base File. It is a temporary subset of the Base File and contains airfields which are potential stopping points between some origin and destination. RF Bases can be created by either the Refuel or the Interplane routines. RF Bases are ranked within the RF Base File by high passenger value and then by low total time. These attributes will be discussed further in the section of this chapter on the Refuel routine.

Information about airfield services and hours of operation is not included in the model. Most airfields that would normally be members of the Base File provide at least the minimum services required during the hours that most travelers request airlift support.

Aircraft. Sabreliners are created by the Read Data routine. The aircrew is modeled as part of the aircraft entity.

"Home station" normally refers to the Sabreliner's base of assignment; however, the model user may employ "home station" to identify any airfield at which he wants the aircraft to terminate its itinerary.

Crew duty start time, maximum crew duty day, and duty day are established by the T-39 Scheduler routine to insure that planned aircraft utilization does not exceed crew duty time limitations.

The number of seats available on each Sabreliner is a variable that is used by both the T-39 Scheduler routine and the Refuel routine.

Every Sabreliner owns an itinerary, which is a set of legs. Each of those legs may own a set of satisfied travel requests.

Travel Requests. Travel requests are created by the Read Data routine. They are initially filed in a set called Unsatisfied Requests. This set is ranked by passenger priority and then by earliest NLT date and time. As they are supported, travel requests are removed from the Unsatisfied Requests set and filed in a set of satisfied requests owned by a leg in the itinerary of the supporting Sabreliner.

Information about the mission request validator (or other contact) is not included among the attributes of travel requests because it is not required to produce a schedule. This information is required by the MAC/DOOF Requirements Branch, and a method for incorporating it into the model will be discussed in Chapter V.

Main

The Main Program simply calls the Read Data, T-39 Scheduler, and Print Schedule routines in sequence. Figure 1 shows the logic flow diagram for the Main Program.

Read Data

The Read Data routine reads from punched cards the airfield, aircraft, and travel request information necessary to prepare a CT-39 initial schedule for a given Julian day. The routine also performs some scheduler functions.

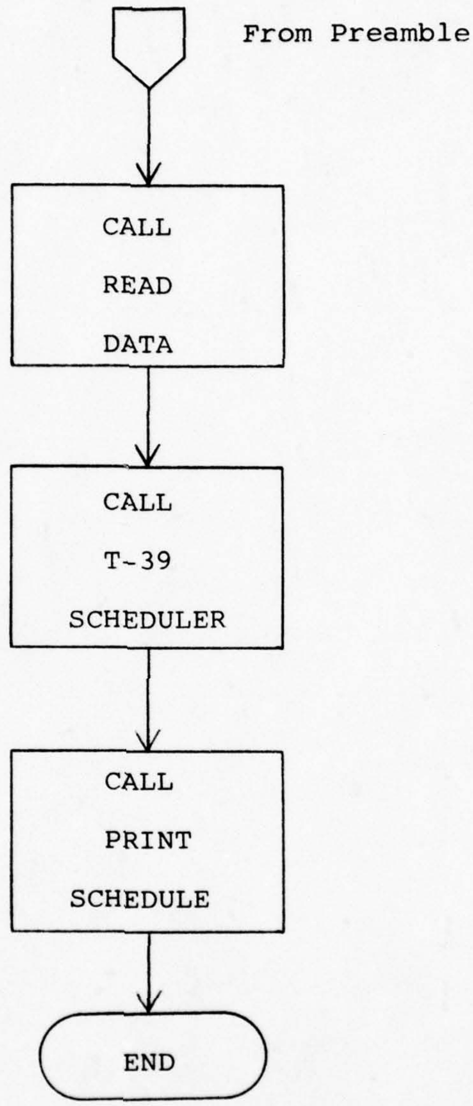


Figure 1. Logic Flow Diagram For Main Program

A logic flow diagram is contained in Figure 2. Specific formats for required inputs are shown in Appendix B.

The Read Data routine identifies and removes from the Unsatisfied Requests set any travel requests with origin or destinations that are not in the Base File set. If a travel request has a passenger load of six, the routine reduces the load to five to align it with the Sabreliner's passenger capacity.

If a priority 1-3 request indicates an inflexible departure or arrival time, the Read Data routine adjusts either the NET or NLT time by the estimated flying time. If a priority 1-3 request has only one minute between the NET time and the NLT time, but neither of these times ends in zero or five, there is probably a mistake in the request. Nevertheless, the routine will make the NET time earlier by subtracting the estimated flying time.

For ease of data processing, the Read Data routine converts all NET and NLT times to hours and hundredths of hours and adjusts all times to a reference time 0.00 hours GMT on the schedule day.

For the scheduler's information, the Read Data routine prints the following items:

a. A list of all airfields in the Base File.

b. The inputs concerning leap year, schedule day, and daylight saving time.

c. The total number of CT-39s available for operational support airlift, and the location and termination

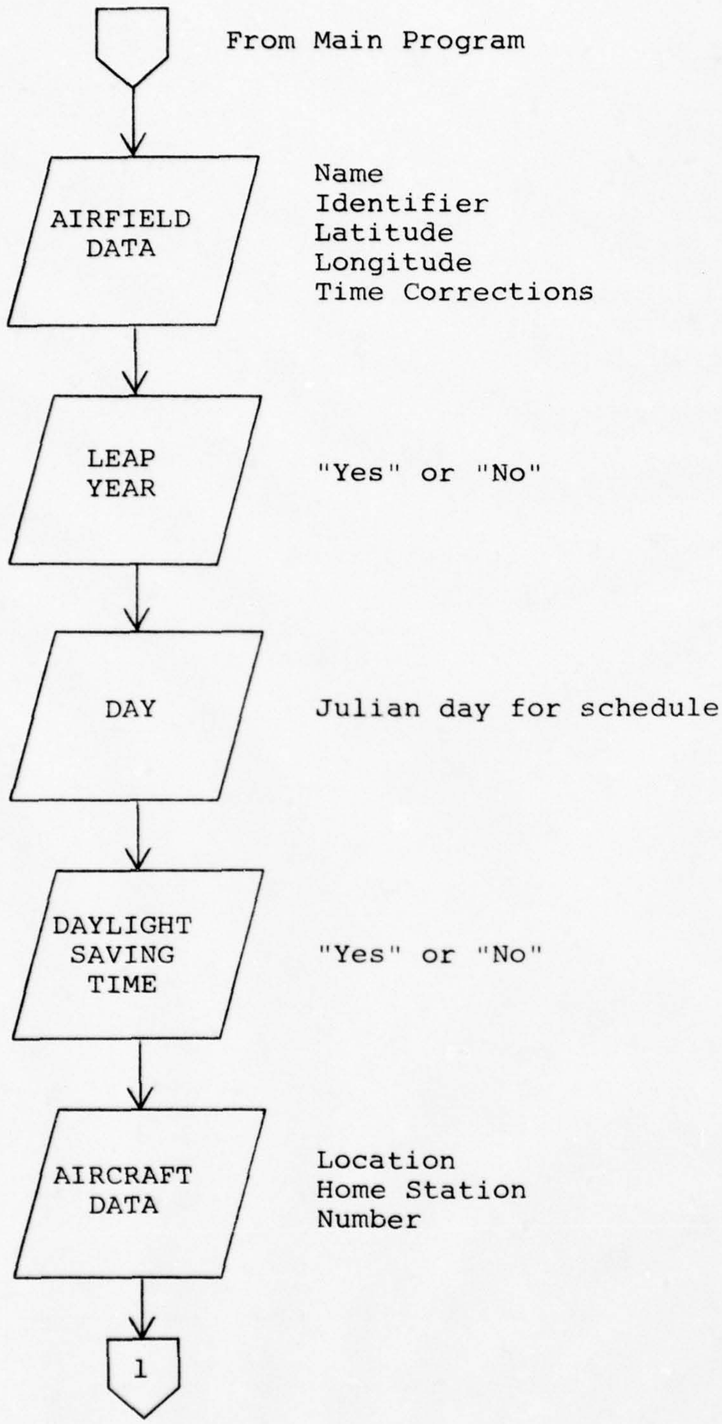


Figure 2. Logic Flow Diagram For Data Routine

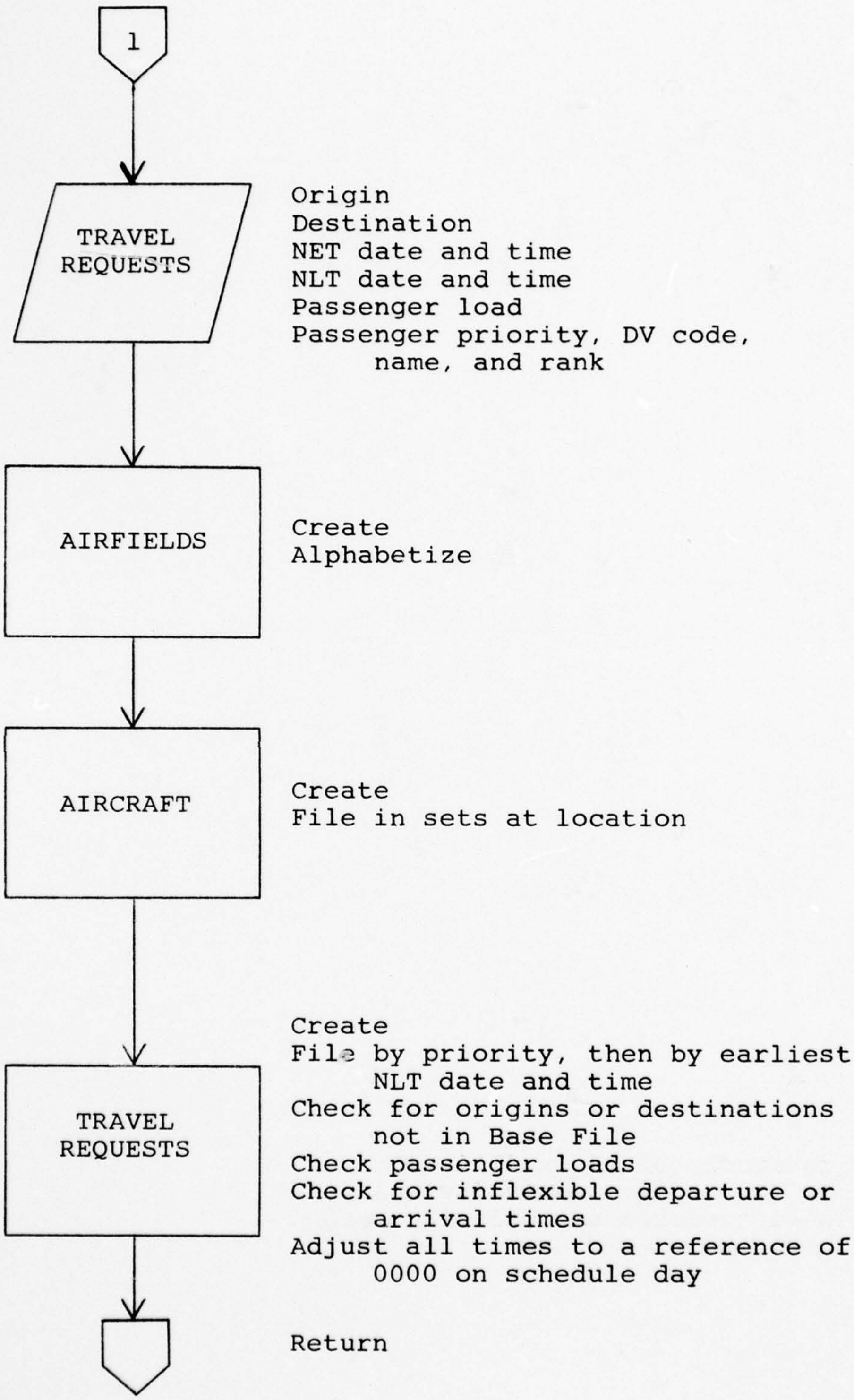


Figure 2. (Continued)

point for each aircraft.

d. A list of all priority 1-3 travel requests with origins and destinations in the Base File (ordered alphbetically by origin identifier).

e. A similar list of all priority 4-12 requests with origins and destinations in the Base File.

f. A prioritized list of all requests that have either an origin or a destination which is not in the Base File.

g. A list of priority 1-3 requests ordered in the sequence that the algorithm will normally consider them for support.

Examples and further explanation of this output are contained in the user's guide in Appendix B.

T-39 Scheduler

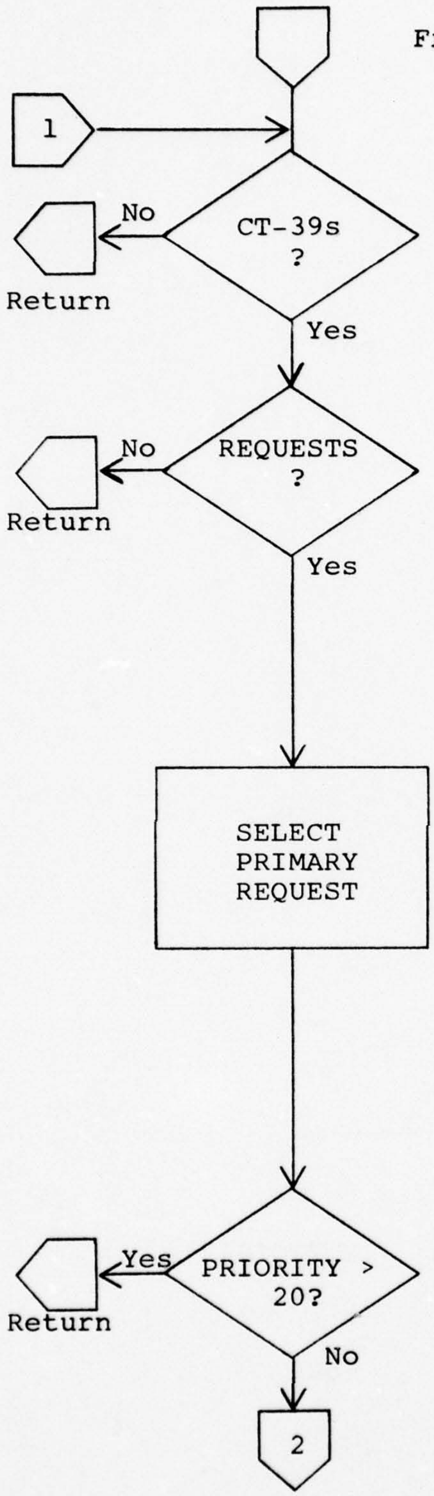
The purpose of the T-39 Scheduler routine is to develop itineraries for the available Sabreliners in a manner that is consistent with the objectives of the model. This section amplifies the logic flow diagram of the routine presented in Figure 3.

The routine creates itineraries one at a time until one of the following conditions is met:

a. All available aircraft have been scheduled.

b. All travel requests have been satisfied.

c. The routine cannot support any of the remaining requests with the aircraft which have not yet been scheduled.



From Main Program

Are there any unscheduled Sabreliners remaining?

Are there any unsatisfied requests remaining?

Selection order from unsatisfied requests:

a. USAF 0-10
b. Scheduler designated
c. Priority 1-3 request with origin and destination same as location and termination point of Sabreliner
d. Prioirty 1-3 interplane request with earliest NLT time
e. Priority 1-3 request with earliest NLT time
f. Highest priority request with passenger load of 5 (then 4, 3, 2, and 1)

If the priority of the primary request is greater than 20, no unsatisfied requests can be supported by the unscheduled Sabreliners remaining.

Figure 3. Logic Flow Diagram For Routine T-39 Scheduler

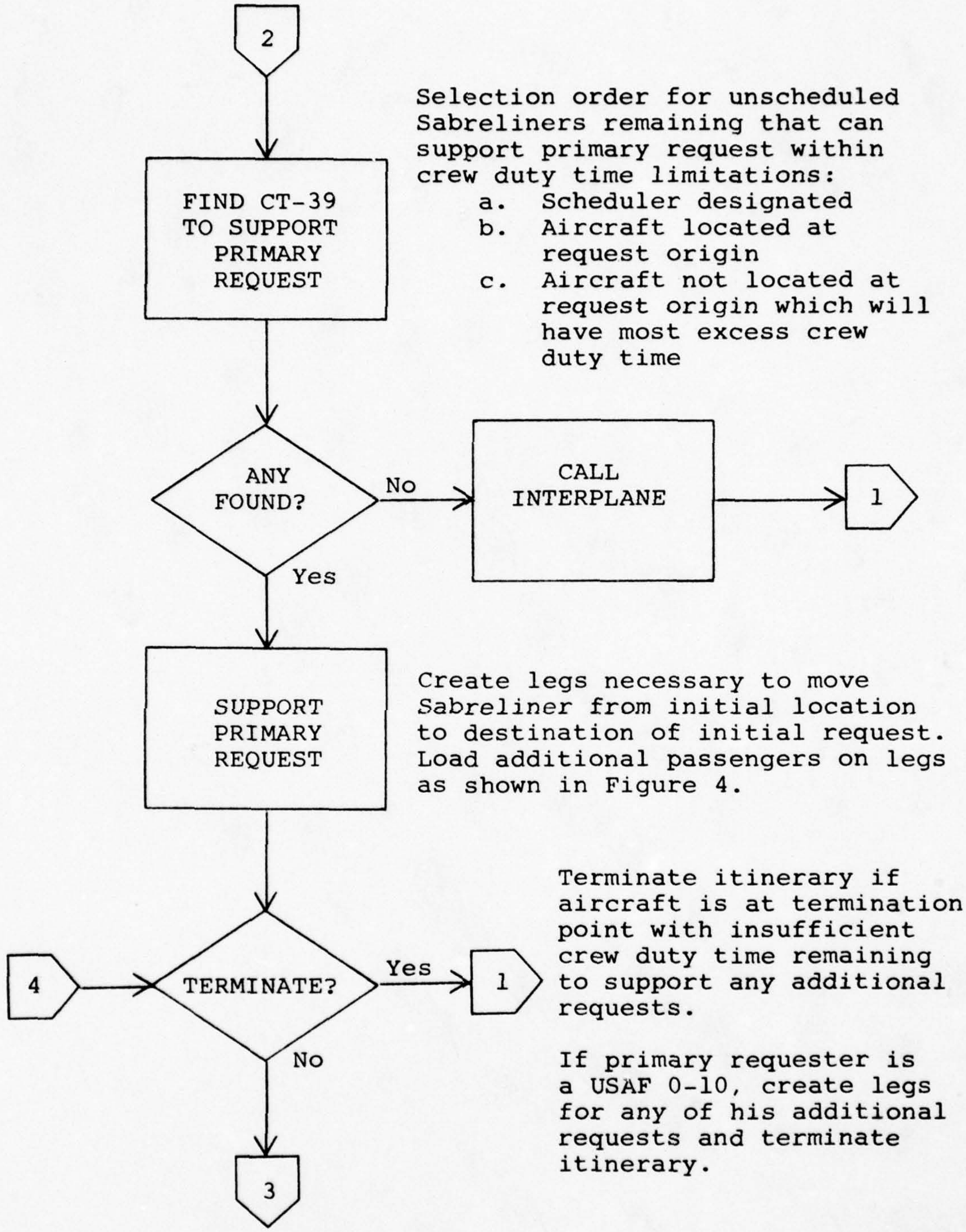


Figure 3. (Continued)

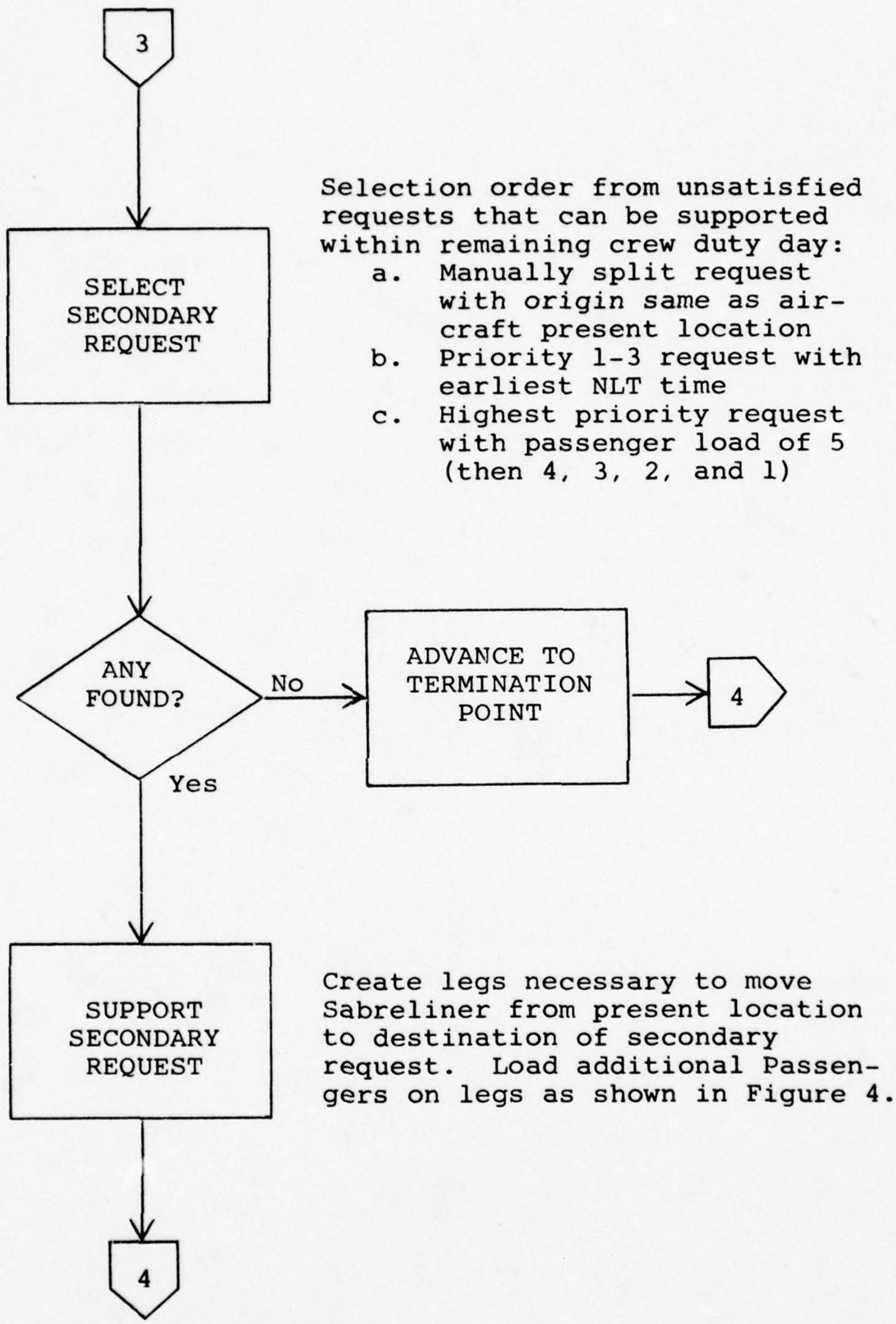


Figure 3. (Continued)

There are three major steps in the development of each itinerary. Each step is governed by a set of heuristic decision rules. First, the routine chooses the primary request to be supported. Second, it finds an unscheduled aircraft to support the request. Third, after all the legs necessary to support the primary request have been created, the routine looks for one or more secondary requests which can be supported within the remaining crew duty time. Each of the major steps has a provision that allows for scheduler intervention.

The first primary requests chosen from the Unsatisfied Requests file are those submitted by Air Force four-star generals. This is a preliminary step to remove special cases from the file. Since most USAF 0-10s receive exclusive use of an aircraft until they complete their travel requirements, a computerized algorithm is not necessary to schedule their missions. The T-39 Scheduler routine simply assigns all requests from the same USAF 0-10 to a single available Sabreliner before considering the other requests.

The next primary requests are those directed by the scheduler. This feature allows the scheduler to evaluate alternative scheduling strategies by changing the order in which primary requests are selected. The methods for accomplishing this and other forms of manual intervention are explained in the user's guide in Appendix B.

A request with a priority greater than 20 is one that has already been selected once as the primary request and

has been found to be unsupportable. When an unsupportable request is selected as the primary request for the second time, none of the unsatisfied requests can be supported by the unscheduled aircraft remaining.

To determine if a particular Sabreliner is able to support the primary request, the T-39 Scheduler routine calls upon the Leg Data routine to compute the travel times from the aircraft location to the request origin, from the origin to the destination, and from the destination to the aircraft termination point. The routine then determines the total travel time based on the assumption that all ground times last one hour. Crew duty start time is calculated based on takeoff from the origin at NET time. From crew duty start time, the maximum allowable crew duty day is determined. If total travel time does not exceed the maximum crew duty day, the aircraft can support the primary request.

Once an aircraft has been selected to support the primary request, no checks are made to insure that the requester arrives at this destination by the NLT time. The routine assumes that if the aircraft departs from the origin at the NET time and proceeds directly to the destination, the requester will arrive on time. To avoid large deviations from a direct route, the routine will not choose a refueling base that adds more than 30 minutes to the great circle route travel time between the origin and the destination.

If no single unscheduled aircraft can support the

request, the Interplane routine is called. If the request is of priority 1-3, the Interplane routine will attempt to divide the request into two separate requests which will both be supported by the remaining unscheduled Sabreliners. If this can be done, the two requests created by Interplane will be the next two primary requests. If it cannot be done, or if the primary request was of priority 4-12, Interplane adds 20 to the priority of the request and files it at the bottom of the Unsatisfied Requests file.

If an aircraft is found to support the primary request, the routine creates the legs necessary to move from the aircraft location to the request origin and from the origin to the destination. If refueling is required along either of these routes (as determined by the Leg Data routine), the Refuel routine is called to create a file of feasible refueling bases. The T-39 Scheduler routine selects the refueling base which has the highest passenger value but which adds no more than 30 minutes to the great circle route travel time. If the Refuel routine does not identify a refueling base, the output of the Print Schedule routine will indicate that manual selection of a refueling base is required.

Because the method for determining if an aircraft can support a primary request within crew duty day limitations is based upon great circle route travel times, a remote possibility exists that the method of selecting a refueling base will cause an aircraft to exceed its maximum crew duty

day. If this happens, the output of the Print Schedule routine will advise the scheduler that manual adjustments to the ground times in the itinerary will probably correct the problem.

Additional passengers are carried on these routes in accordance with the passenger loading scheme illustrated in Figure 4. A limitation of the passenger loading scheme is that it only considers one major route segment at a time. For example, if the aircraft supporting the primary request is not located at the origin of the request:

a. Passengers traveling between the aircraft location and the request origin will be loaded.

b. Passengers traveling between the origin and the destination will be loaded.

c. Passengers traveling between the aircraft location and the request destination will not be loaded.

If the Refuel routine is called to find a refueling base along a major route segment between points A and B, passengers may be loaded if they are traveling:

a. From A to B.

b. From A to the refueling base.

c. From the refueling base to B.

After the aircraft has advanced to the destination of the primary request, the routine determines if the itinerary should be terminated. If the itinerary is not to be terminated, the routine searches for secondary requests which can be supported in the remaining crew duty time. The

Case 1: No Refueling Required

Origin
Departure Time = DTA

Destination
Arrival Time = ATB

Load passengers from selected request
Determine number of seats available

Until all seats are filled
or
Until all requests have been searched

Find the highest priority unsatisfied request to travel from A to B with:

NET time $\leq$ DTA

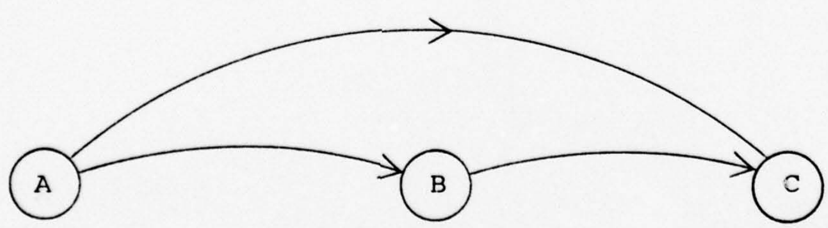
NLT time $\geq$ ATB

Passenger load $\leq$ seats available

If found:

Load passengers
Reduce number of seats available
Continue search

Case 2: Refueling Required



Origin
Departure Time = DTA

Refueling Base
Arrival Time = ATB
Departure Time = DTB

Destination
Arrival Time = ATC

(Continued on next page)

Figure 4. Passenger Loading Procedures

Load passengers from selected request
Determine number of seats available from A to B, B to C, and A to C

Until all seats are filled or until all requests have been searched
Find the highest priority unsatisfied request to:

Travel from A to B with:

NET time $\leq$ DTA

NLT time $\geq$ ATB

Passenger load $\leq$ seats available from A to B

Or

Travel from B to C with:

NET time $\leq$ DTB

NLT time $\geq$ ATC

Passenger load $\leq$ seats available from B to C

Or

Travel from A to C with:

NET time $\leq$ DTA

NLT time $\geq$ ATC

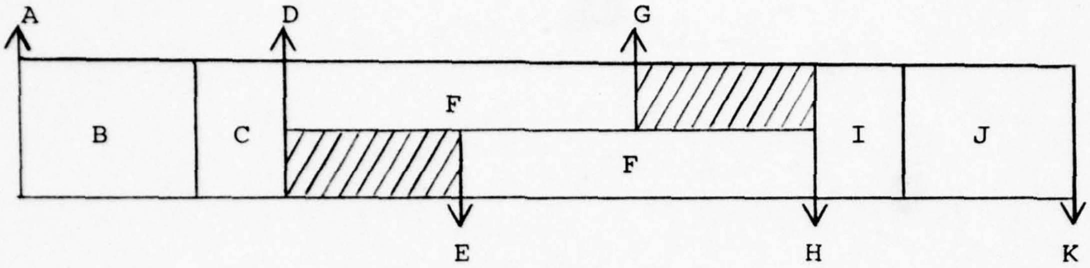
Passenger load $\leq$ seats available from A to C

If any of the above are found

Load Passengers
Reduce seats available from A to B, B to C, and A to C as appropriate

Continue search

Figure 4. (Continued)



A = Earliest possible takeoff from present location
(A = arrival time + 1 hour)

B = Travel time to request origin
(B = 0 if origin is present location)

C = Ground time at request origin
(C = 0 if origin is present location, C = 1 hour otherwise)

D = Earliest possible takeoff from request origin
(D = A if origin is present location)

E = Latest allowable takeoff from request origin
(E = H - F)

F = Travel time from origin to destination

G = Earliest possible landing at destination
(G = D + F)

H = Latest allowable landing at destination
(H = K - I - J)

I = Ground time at request destination
(I = 0 if destination is aircraft termination point, I = 1 hour otherwise)

J = Travel time from destination to aircraft termination point (J = 0 if destination is aircraft termination point)

K = Latest allowable landing at termination point
(K = Crew duty start time + maximum crew duty day, K = H if destination is aircraft termination point)

Secondary request can be supported if:

$$\text{NET time} \le E$$
$$\text{NLT time} \ge G$$
$$\text{and } H - \text{MAX(NET time, D)} \ge F$$

Figure 5. Secondary Request Selection Procedures

method for determining if a request can be supported within the remaining crew duty day is illustrated in Figure 5. After a secondary request is selected, procedures are similar to those used for the primary request. The only major difference is that the takeoff from the request origin will be at the later of the earliest possible takeoff time or the NET time of the request.

To assist the scheduler, this routine displays a list of the order in which primary requests were supported and the location and termination point of the Sabreliner selected to support each primary request.

Leg Data

The basic purpose of the Leg Data routine is to estimate CT-39 travel time between a given origin and destination. The following assumptions were made in this routine:

a. The earth is a perfect sphere.

b. The schedule will only consider points in the northern and western hemispheres.

c. Sabreliners always travel a great circle route between two points.

d. Cruise true airspeed is always 400 knots.

e. The wind is directly from the west at either 25 knots or 65 knots, depending on the season.

f. Cruise time equals the great circle distance from origin to destination divided by the estimated ground speed.

g. Additional flying time for climb, descent, approach, and landing can be represented by a constant.

h. Total flying time between two points will not exceed 3 hours and 15 minutes.

i. Fuel consumption is 1900 pounds per hour of total flying time.

The assumptions above imply that if the cruise time between two points exceeds some constant value, refueling will be required. The Leg Data routine estimates total travel time between two points by adding to total flying time one hour of ground time for each refueling required.

A simplified logic flow diagram is shown in Figure 6.

The assumptions in this routine were made in order to obtain a reasonable approximation of the flying time estimates used by the MAC/DOOF Planning Branch. Table VI compares output from the Leg Data routine with MAC CT-39 flying time estimates. The estimates differ by less than five minutes in all cases and by less than two minutes in most cases.

MAC/DOOF has obtained satisfactory results using their flying time estimates listed in Table VI. Based on our experience as aircrew members, we feel that the differences between the output of the Leg Data routine and the MAC estimates are not significant for purposes of developing the initial schedule.

We also feel that the Leg Data routine produces a reasonable estimate of CT-39 fuel consumption; however,

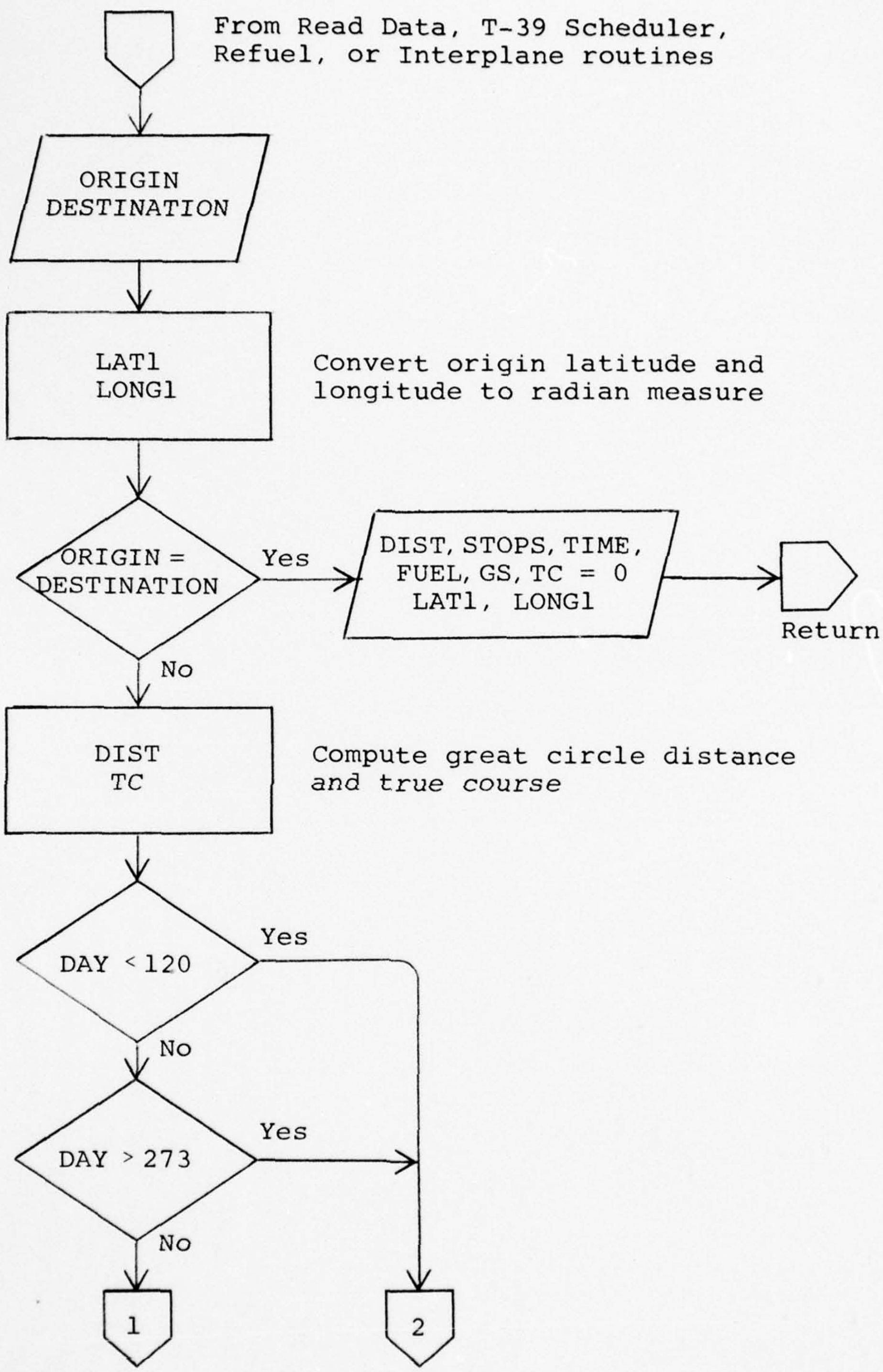


Figure 6. Logic Flow Diagram for Leg Data Routine

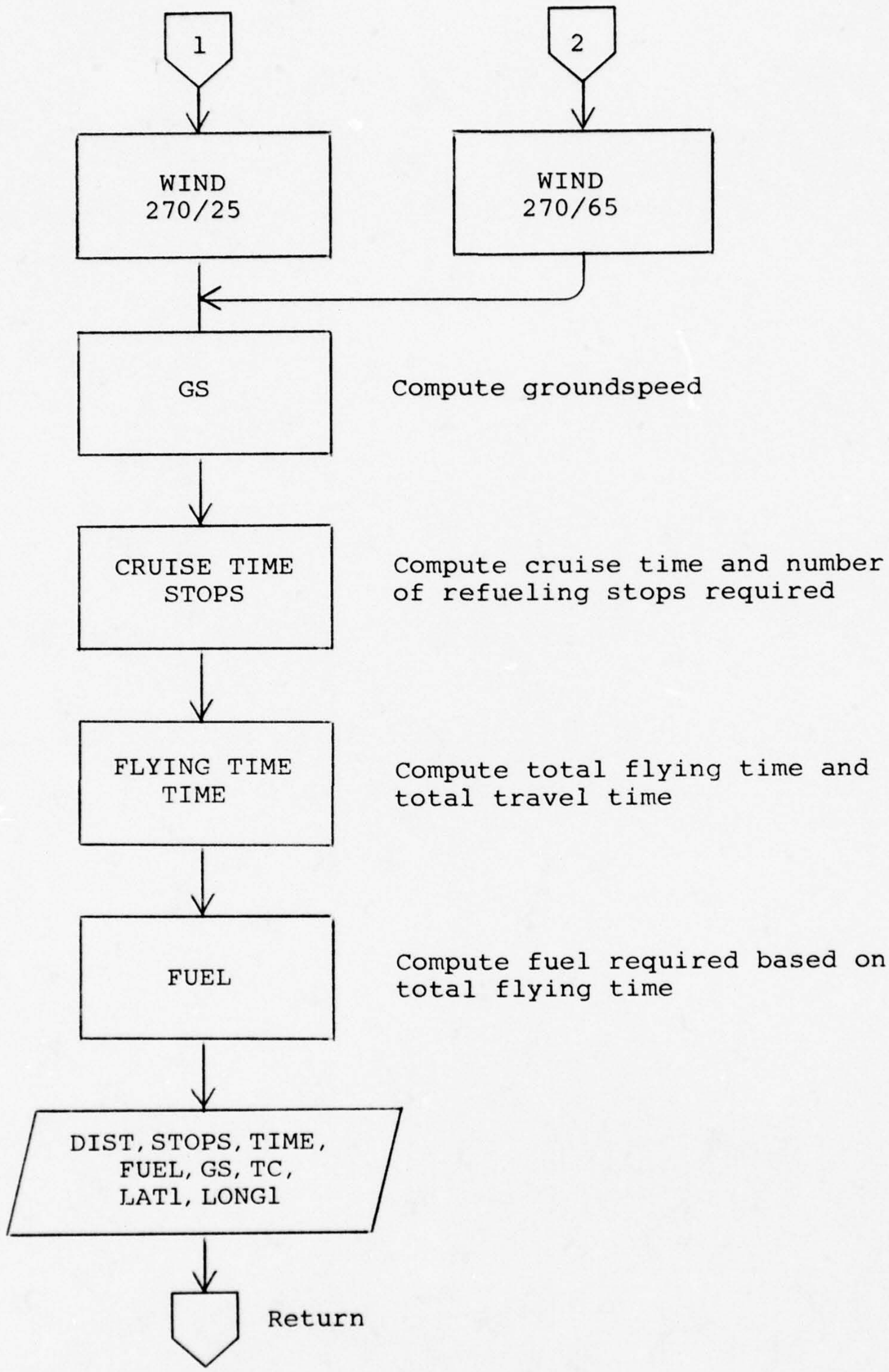


Figure 6. (Continued)

Table VI

Comparison of Output From Leg Data Routine

With MAC/DOOF CT-39 Flying Time Estimates

WIND (DEGREES/ KNOTS)	DISTANCE (NAUTICAL MILES)	TRUE COURSE (DEGREES)	MAC/DOOF ESTIMATE (HOURS)	LEG DATA ESTIMATE (HOURS)	DIFFERENCE (MINUTES)
270/25	275	270	1.0	1.05	+2.7
	435		1.5	1.47	-1.7
	625		2.0	1.98	-1.2
	815		2.5	2.49	-0.9
	1000		3.0	2.98	-1.2
	315	090	1.0	1.05	+3.2
	495		1.5	1.48	-1.4
	710		2.0	1.98	-1.0
	930		2.5	2.50	0.0
	1145		3.0	3.01	+0.4
	300	180/360	1.0	1.06	+3.8
	465		1.5	1.48	-1.4
	675		2.0	2.00	+0.2
	875		2.5	2.50	+0.2
	1075		3.0	3.00	+0.3
270/65	245	270	1.0	1.04	+2.6
	395		1.5	1.49	-0.5
	570		2.0	2.01	+0.8
	745		2.5	2.54	+2.2
	915		3.0	3.04	+2.6
	345	090	1.0	1.05	+3.3
	530		1.5	1.45	-2.9
	770		2.0	1.97	-1.9
	1000		2.5	2.46	-2.2
	1240		3.0	2.98	-1.2
	300	180/360	1.0	1.07	+4.4
	465		1.5	1.49	-0.6
	675		2.0	2.02	+1.4
	875		2.5	2.53	+1.7
	1075		3.0	3.04	+2.2

we do not have adequate data to confirm our opinions. This is an area for further investigation and will be discussed in Chapter V.

Refuel

The purpose of the Refuel routine is to create a set of airfields which are feasible refueling stops for a Sabreliner flying between two points. The T-39 Scheduler routine will call the Refuel routine when a Sabreliner must refuel (as determined by the Leg Data routine) to travel from some origin to some destination. The logic flow diagram in Figure 7 explains how the Refuel routine identifies feasible airfields.

For computational efficiency, the routine constructs a search region prior to evaluating airfields for feasibility. This region reduces the number of bases which must be evaluated by excluding from consideration bases which obviously would not be selected. Figure 8 illustrates the manner in which the search region is constructed.

For each airfield discovered to be feasible, an RF Base is created. The Refuel routine then follows T-39 Scheduler decision logic to determine which additional travel requests will be supported if that RF Base is selected as the refueling point. Each such request is assigned a passenger value, and the passenger value for the RF Base is the sum of the passenger values of all additional requests that will be supported by selecting that base.

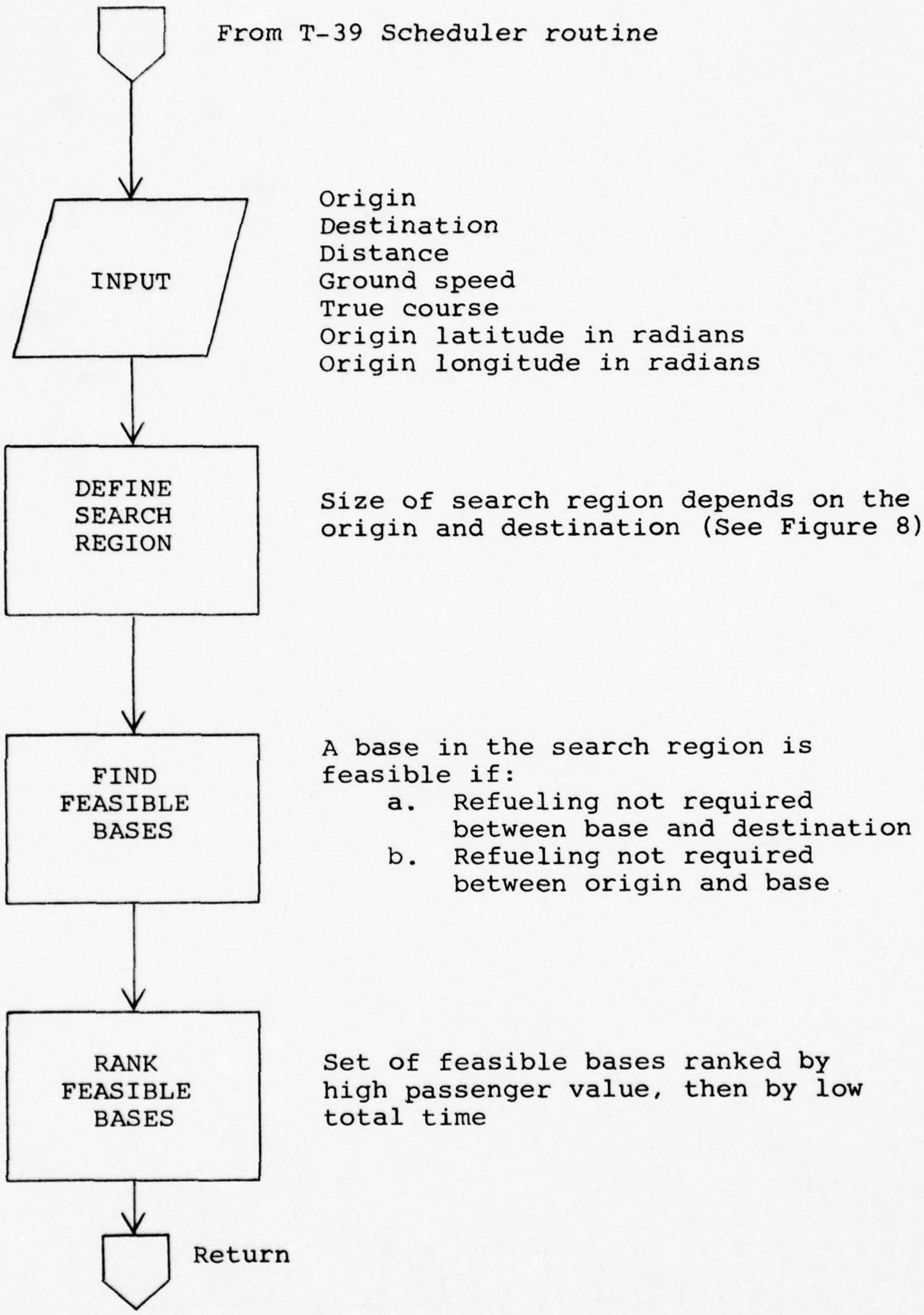
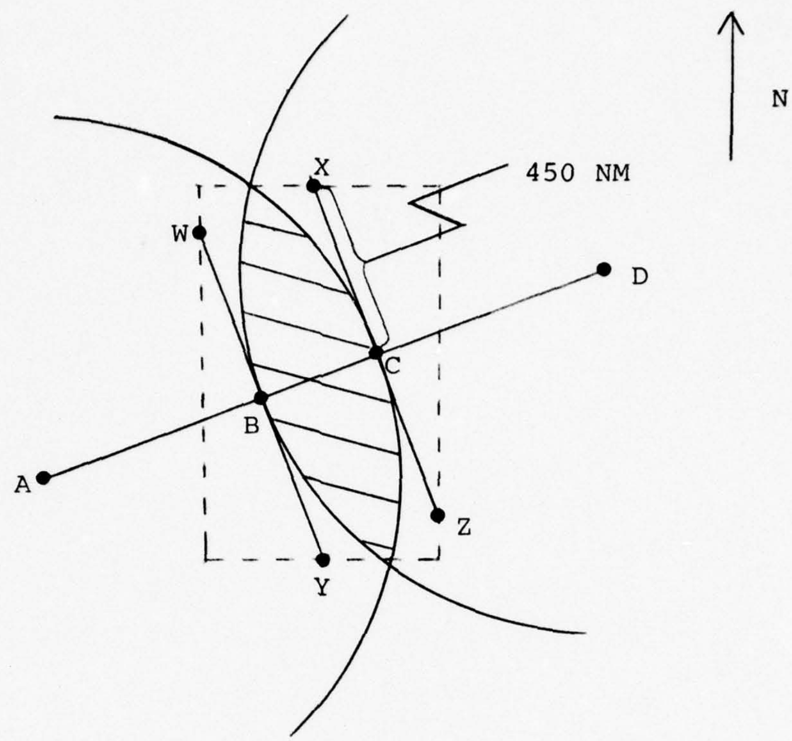


Figure 7. Logic Flow Diagram for Refuel Routine



Area containing feasible refueling bases between A and D

- - - Search region boundaries

A Origin

B Farthest point along true course from A to D from which D can be reached without refueling

C Farthest point along true course from A to D that can be reached from A without refueling

D Destination

W
X
Y
Z
Points located by the position routine. Extreme north, south, east, and west coordinates of these points bound the search region.

Figure 8. Search Region Construction by Refuel Routine

The passenger value scheme assumes that no more than nine additional requests will be supported over any two legs, and it assures the following:

a. A route that will support a priority 1-3 request will be preferred to one that will not.

b. Among routes that will support priority 1-3 requests, the route that will support the highest priority request will be preferred.

c. Among routes that will support priority 1-3 requests, if the routes will support only requests of the same priority, the route that will support the largest number of requests will be preferred.

d. When all other factors are equal, the route that will support the largest number of passengers will be preferred.

The passenger value for a travel request is assigned in the following manner:

$$PV = 10^{(4 - PRI)} + PAX \times (1 + 1/(PRI + 9)) \qquad (1)$$

if the request priority is 1-3, or

$$PV = PAX \times (1 + 1/(PRI + 9)) \qquad (2)$$

if the request priority is 4-12,

where

PV is the passenger value for the travel request

PRI is the priority of the travel request

PAX is the passenger load of the travel request

Position

The Position routine applies formulas from spherical trigonometry to determine coordinates of points used in defining the search region of the Refuel routine.

Interplane

The purpose of the Interplane routine is to break a request that cannot be supported by any of the remaining unscheduled Sabreliners into two requests, each of which will be supported. Figure 9 contains a logic flow diagram of this routine.

Figure 10 illustrates how a search region is constructed to reduce the number of airfields which are evaluated as potential interplane bases. The Interplane routine follows T-39 Scheduler aircraft selection logic to determine if an airfield is a feasible interplane base.

An RF Base is created for each airfield found to be feasible. Each RF Base is assigned a passenger value based on the additional travel requests that can be supported if that airfield is selected as the interplane base. The total time for the RF Base is the time required to travel from the request origin to the request destination.

If any feasible bases are found, the one with the highest passenger value will be selected as the interplane base. If more than one RF Base has the same passenger value, the one with the lowest total time will be preferred.

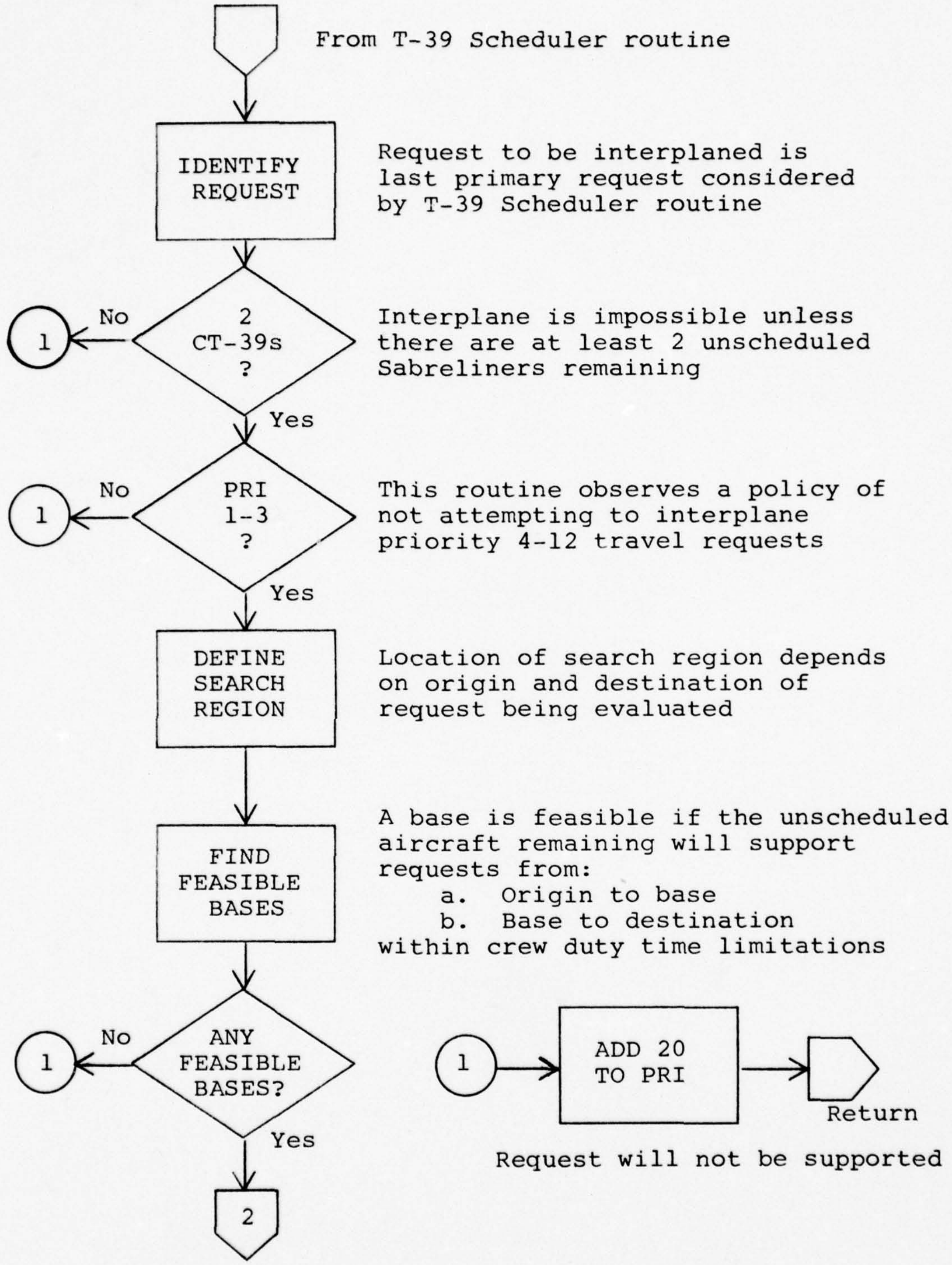


Figure 9. Logic Flow Diagram for Interplane Routine

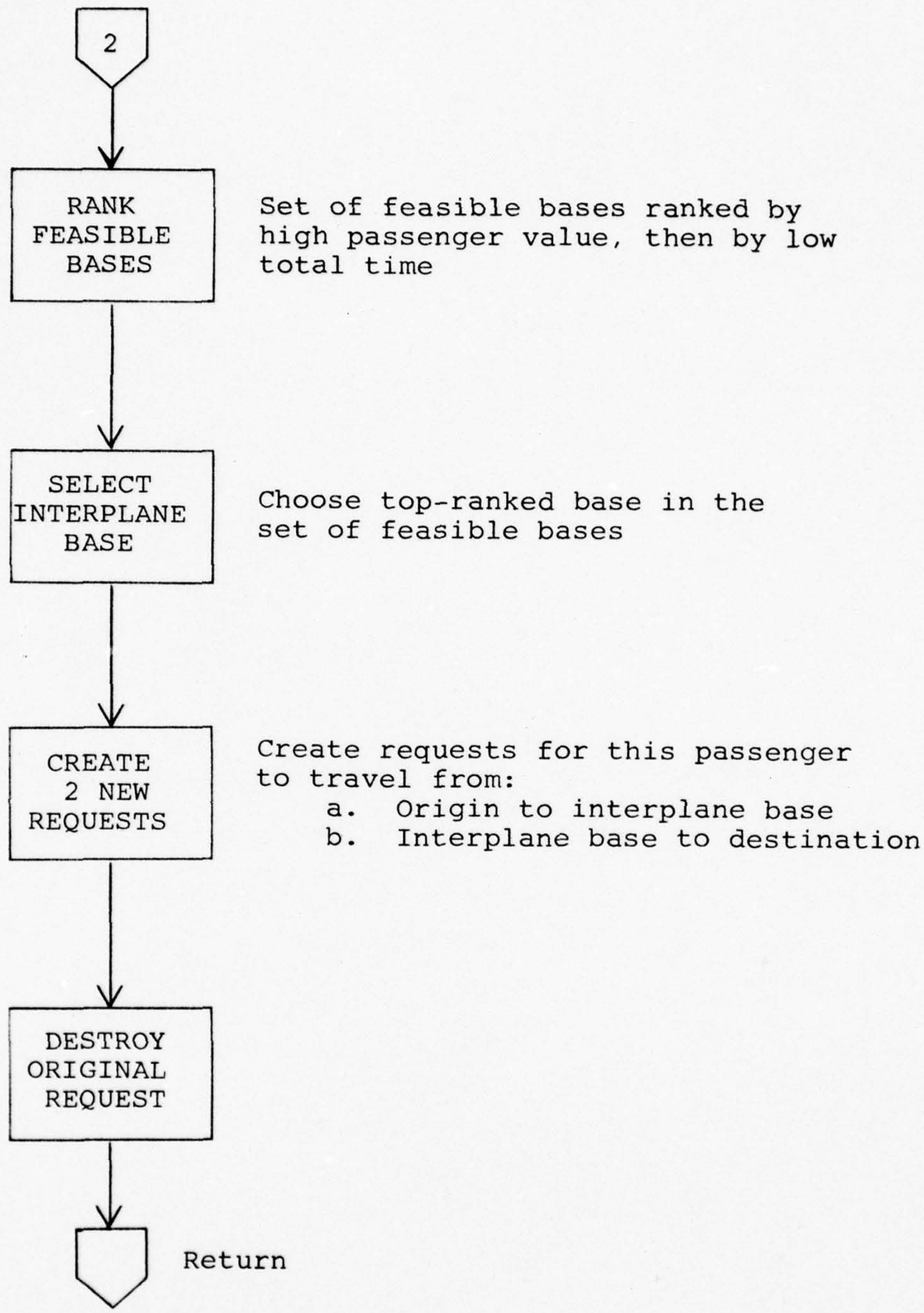
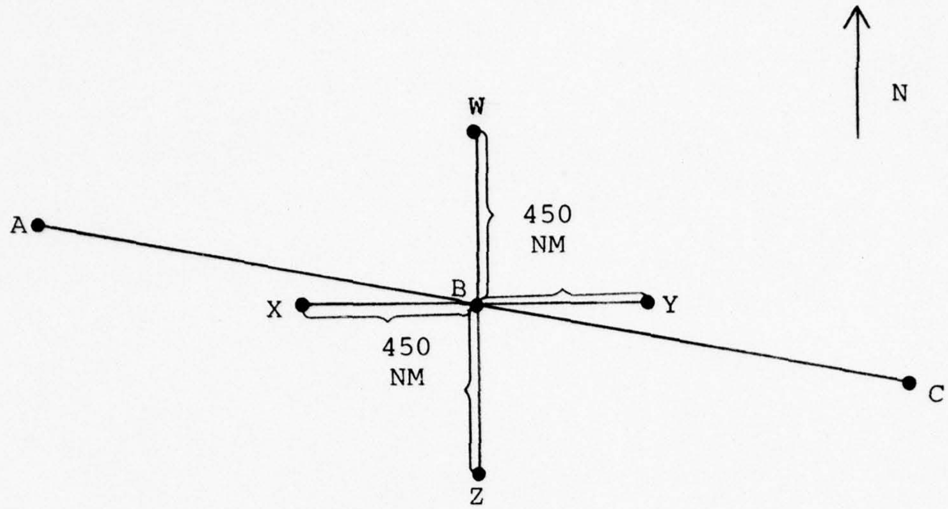


Figure 9. (Continued)



A Origin

B Midpoint between A and C

C Destination

If there are no unscheduled Sabreliners identified to terminate their itineraries at points other than their initial locations:

> The search region is bounded by the extreme north, south, east, and west coordinates of points
>
> W, X, Y, and Z

Otherwise:

> The search region is bounded by the extreme north, south, east, and west coordinates of points
>
> A, C, W, X, Y, and Z

Figure 10. Search Region Construction by Interplane Routine

When two new requests are created, the NET time for the first request is the traveler's original NET time. The NET time for the second request is within 30 minutes to an hour of the traveler's planned arrival at the interplane base. Any requests created in this manner will be supported by the T-39 Scheduler routine.

Print Schedule

This routine prints the schedule developed by the T-39 Scheduler routine. For each available aircraft at a particular base, the routine prints this information:

a. Crew duty start time (both in Julian date and GMT and in local date and time).

b. A complete itinerary. For each leg of the itinerary, the routine lists the origin and departure time, the destination and arrival time, and the passengers to be carried.

c. A list of all unsupported requests for travel along the aircraft's route of flight. These requests may not have been supported for the following reasons:

1. NET or NLT times were not compatible with the itinerary.

2. Passenger loads were too large for the number of seats available.

3. The desired origin and destination were not considered as a pair by the T-39 scheduler routine.

This list of unsupported requests assists the scheduler

in recommending to travelers revised dates or times that might permit their requests to be supported.

After the itineraries for all available aircraft have been printed, the routine publishes two additional lists to assist the scheduler. The first is a list of all unsupported requests with NLT times that will not allow them to be supported in the next schedule day. The second list contains all the unsupported requests that may be supported in the next schedule day. Both lists have two parts, each of which is alphabetized by origin identifier and then by destination identifier. The first part of each list contains only priority 1-3 requests, while the second part contains the priority 4-12 requests. These lists are designed for ease of use in making manual adjustments to the schedule, and their formats are similar to computer lists currently used by MAC/DOOF in their manual scheduling process.

Examples and explanations of output from this routine are included in Appendix B.

Summary

This chapter has discussed the objectives of the model, the basis for the scheduling algorithm, and the routines of the model. Chapter IV examines model performance and evaluates model validity.

IV. RESULTS

This chapter evaluates the validity of the model by comparing schedules produced using the model to those produced by MAC/DOOF. Addressed first is how the necessary data were gathered. Next, four schedules prepared using the model are compared with the corresponding four schedules developed by the MAC/DOOF Planning Branch. Finally, limitations of the methods of comparison are discussed.

Data Collection

MAC/DOOF provided their manually prepared schedules and the travel requests from which they were developed for the following days:

a. Thursday, 7 September 1978 (Julian day 250).

b. Saturday, 6 January 1979 (Julian day 006).

c. Sunday, 7 January 1979 (Julian day 007).

d. Monday, 8 January 1979 (Julian day 008).

The researchers used the model to develop four initial schedules from the travel requests provided. Schedules for days 250 and 006 considered all travel requests; schedules for days 007 and 008 considered only priority 1-3 requests.

The general approach for creating a schedule with the model is outlined in the user's guide in Appendix B. A more detailed description of how the researchers prepared the schedule for day 250 is presented in Appendix C.

The schedules for days 250 and 008 took the most time

to complete. Each took from 5 to 10 runs of the model. Normal execution times for the model were less than two minutes on a CDC 6600 computer. Because the researchers prepared their schedules by submitting batch jobs, each schedule required a considerable amount of time to complete. However, if the model were adapted for interactive use from a remote terminal, schedule preparation that requires 5 to 10 runs of the model could be accomplished in less than two hours. Schedules that require a large number of adjustments to travel times could conceivably take longer.

Comparison of Schedules

Both the schedules prepared manually by MAC/DOOF and the schedules prepared by the researchers supported all priority 1-3 requests. Since both methods met the first criterion for a good schedule, the basis for comparing the schedules for days 250 and 006 was the total number of passengers supported, and the basis for comparing the schedules for days 007 and 008 was the number of aircraft used to support the priority 1-3 requests.

Data from the comparison are shown in Table VII. In all cases, the schedules produced by using the model improved upon the manually-developed schedules either by supporting more passengers or by supporting the same set of travel requests with fewer Sabreliners.

The computerized model has demonstrated a methodology that will enable MAC/DOOF personnel to quickly produce a

TABLE VII

Comparison of MAC/DOOF Schedules With Model Schedules [a]

Day	Pri 1-3	Pri 4-12	Total Pax Demand	CT-39s Available	Scheduled by DOOF CT-39s	Scheduled by DOOF Pax	Scheduled with Model CT-39s	Scheduled with Model Pax	Improvement
250	54	242	605	39	39	266[b] (272)	39	290	9% More Pax
006	8	58	121	13	13	65[b] (66)	13	71	9% More Pax
007[c]	20	-	-	23	19	-	18	-	6% fewer CT-39s
008[c]	57	-	-	39	38	-	37	-	3% fewer CT-39s

a. Schedules were prepared in different environments. See text for explanation.

b. Top number is total passengers that could have been carried if all Sabreliners were limited to five seats. (Model considers all Sabreliners to have only five seats). Bottom number is total passengers actually scheduled by MAC/DOOF.

c. Priority 4-12 requests were not scheduled.

good CT-39 initial schedule. However, this fact alone does not present a complete picture of the situation.

Limitations of Comparison

The comparison of the two scheduling methods is misleading without further explanation. Table VII does not consider the different environments in which the schedules were created.

MAC/DOOF schedulers work in a dynamic environment. They estimate that 30 percent of all priority 1-3 requests are either received or cancelled after the initial scheduling has begun. Appendix A shows how the deadline for submission of priority 3 requests overlaps the normal initial schedule preparation sequence. Based on the results of an informal study, MAC/DOOF personnel contend that they could improve their effectiveness by 10 percent if all requests were on hand at the start of the scheduling process (Ref 10). While the specifics of the study and its results are somewhat uncertain, the conclusion that Planning Branch personnel could improve their effectiveness seems reasonable.

In contrast to MAC/DOOF, the researchers worked in a static environment. All travel requests were on hand prior to beginning schedule development. The improvement achieved over the manually-prepared schedules seems to support the intuitive conclusion reached by MAC/DOOF. The model helped to produce improved schedules not because it was greatly superior to the Planning Branch schedulers but because it

had more complete information at the start of the scheduling process.

Regardless of the amount of information on hand, the model extends a scheduler's capabilities by allowing the exploration of many more scheduling alternatives than are possible to consider under the current manual system. Clearly, though, the greatest single benefit that the model could offer MAC/DOOF would be the opportunity to postpone the start of their initial schedule preparation until closer to the deadline for submission of priority 3 requests. Because of the speed with which it executes, the model should permit such a delay.

Further research is necessary to answer these two questions that are significant to MAC:

a. How long can the start of schedule preparation be delayed?

b. Would such a delay result in sufficient schedule improvement to offset the cost of adopting and using the model?

Chapter V will present conclusions and recommendations. Foremost will be the recommendation that MAC/DOOF gather the information needed to answer the questions above.

V. CONCLUSIONS AND RECOMMENDATIONS

This chapter assesses the extent to which we have achieved our objectives and offers some recommendations regarding the future of the model we have developed.

Summary and Conclusions

Chapter I stated the primary objective was to develop a computerized model that demonstrates a methodology which will enable MAC/DOOF personnel to quickly prepare a good CT-39 initial schedule. In pursuit of this objective, these major steps were accomplished:

a. The essential elements of the CT-39 operational support airlift mission preparation process were identified.

b. The characteristics of a good schedule were identified.

c. A heuristic scheduling algorithm was developed and incorporated into a computerized scheduling model.

d. Schedules produced using the model were compared to actual schedules produced by the MAC/DOOF Planning Branch.

We have demonstrated that a computerized scheduling model can be developed. This model cannot replace a human scheduler, but it can extend a scheduler's capabilities.

We have also demonstrated that the model can produce good operational support airlift mission initial schedules, even when used by personnel with little experience in this

type of scheduling. The primary advantage that the model offers over a manual system is the speed with which the schedule can be completed. It also permits the user to quickly evaluate alternative scheduling strategies which probably would not even be considered if the schedule were being prepared manually.

To be of greatest value to MAC/DOOF, the model must allow the Planning Branch to delay the start of their initial schedule preparation process until closer to the deadline for submission of priority 3 requests. Additionally, the model would have to be modified for interactive use from a remote terminal. Recommendations on accomplishing this modification are included below.

Recommendations

We recommend that MAC/DOOF conduct a benefit-cost analysis to determine if adoption of this scheduling model would be desirable. The cost of implementing this model could be obtained by submitting a data processing request to the MAC Office of Command Data Automation. This office has systems analysts who can determine how MAC/DOOF can obtain access to the MAC SIMSCRIPT II.5 compiler. The Simulation Analysis Branch of this office has personnel who are familiar with SIMSCRIPT II.5 and who could help make the programming modifications necessary to allow MAC/DOOF to use the model from a remote terminal. The benefits of implementation would be more difficult to quantify. The primary factor in determining

the benefits would be the length of time MAC/DOOF could delay the start of schedule preparation.

If the results of the analysis support implementation, these areas of the model should be changed to make it compatible with the MAC/DOOF Planning Branch operation:

a. The Read Data routine should be modified to read both airfield data and travel requests from a permanent file. The travel requests are already on a file in the current MAC/DOOF computer system. The current system also has a filter which insures that only the requests for the schedule day of interest are read.

b. The name and telephone number of the mission request validator or other contact should be added to the attributes of each travel request. This additional information would necessitate changing only a few statements in the Preamble, the Read Data routine, and the Print Schedule routine.

c. The Read Data routine should be modified to allow input of aircraft availability from a remote terminal. The T-39 Scheduler routine should be modified to permit all forms of manual intervention and data manipulation described in the user's guide to be accomplished from a remote terminal.

If the model is implemented, there is one area we recommend for further investigation. That area is the verification and validation of the fuel consumption figures computed in the Leg Data routine. If these figures prove to be sufficiently accurate, the model could be modified to track the

fuel remaining for each Sabreliner as it progresses along its itinerary. The fuel remaining after each leg could be compared to the fuel required for the next leg, and if that fuel were available, the ground time between legs could be reduced to 30 minutes within the model.

If the model is not adopted for scheduling operational support airlift missions, it can still be used as a policy evaluation tool. For example, if MAC were to decide to reduce the number of operating locations for the CT-39 fleet, the model could assist in determining the best locations for their Sabreliner detachments. Repeated runs of the model could be made using different aircraft location strategies in an attempt to support travel requests for several typical days chosen at random. Aircraft location strategies could then be compared on the basis of their abilities to support typical demands for the use of CT-39 resources. The model developed here is, most basically, a representation of the way MAC uses its CT-39 fleet. As such, it may be helpful in examining numerous issues associated with the use of that resource.

Bibliography

1. AFR 60-23. Operational Support Aircraft Management. Washington: Department of the Air Force, October 21, 1977.

2. Griesser, Major Thomas, USAF. Former Chief, Data Analysis Section, HQ MAC Administrative Airlift Division (telephone conversation). Scott AFB, IL, March 1, 1978.

3. Hillier, J. M., and G. S. Lieberman. Operations Research (Second Edition). San Francisco: Holden-Day, Inc., 1974.

4. Hoeber, Francis P. Casebook: Military Applications of Modeling (Draft). McLean, VA: General Research Corporation, December 1978.

5. Janes, Colonel Norman, USAF. Chief, HQ MAC Administrative Airlift Division (personal interview). Scott AFB, IL, September 6, 1978.

6. MacDougall, Major Dougall, USAF. HW MAC Administrative Airlift Division (personal interview). Scott AFB, IL, September 6, 1978.

7. MACR 55-1, Volume I. Airlift Operations. Scott AFB, IL: Military Airlift Command, January 15, 1976.

8. MACR 55-1, Volume IV. Administrative Airlift Operations. Scott AFB, IL: Military Airlift Command, August 27, 1976.

9. Overacker, Colonel William E., USAF. Operational Support Airlift Summary for April-June 1978. Scott AFB, IL: HQ MAC Deputy Chief of Staff for Operations, August 1978.

10. Owings, Major Will L., USAF. Chief, Planning Branch, HQ MAC Administrative Airlift Division (personal interview and telephone conversations). Scott AFB, IL, September 6, 1978-February 21, 1979.

11. Sherman, M. N. An Automated Planning Technique for the Optimization of Airlift Mission Route Selection. Hanscom Field, MA: AFSC Electronic Systems Division, Directorate of Planning and Technology, April 1972. (AD 868 278).

12. Young, S. H. H. "Gallery of USAF Weapons," Air Force Magazine, 61 (5): 114-131 (May 1978).

APPENDIX A

MAC/DOOF Initial Schedule Preparation Timetable

Below is the timetable MAC/DOOF Planning Branch follows when preparing an initial schedule for Friday during a normal five-day work week. The other mission day timetables are similar except for some compression for non-duty days.

Deadline for submitting Priority 4-12 requests.	MONDAY < 2400	
	TUESDAY	
	0700 >	Begin preparing initial schedule.
	1730 >	Complete draft of feasible initial schedule.
Deadline for submitting Priority 3 requests.	< 2400	
	WEDNESDAY	
	0730 >	Update feasible schedule for all cancellations and add new priority 3 requests. Add Priority 4-12 requests as time allows.
	1000 >	Submit completed initial schedule to Requirements Branch.
	THURSDAY	Change Section updates schedule as required.
	FRIDAY	Fly

APPENDIX B

User's Guide

The user's guide for the model consists of three sections. The first section specifies the formats for all inputs. The second section contains samples of output and explains how to interpret them. The third section describes a technique for using the model.

Section I. Input Formats

This section identifies all required and optional Hollerith card data inputs. The cards must be placed in the data deck in the listed sequence.

Airfield Data

Item	Card Column	Note
Name	01-25	
Identifier	26-29	For bases with only 3-letter FAA identifiers, precede the identifier with the letter "K" (Example: enter KBKF for Buckley ANGB)
Latitude	31-35	Must be entered in degrees and hundredths with decimal in column 33
Longitude	37-42	Must be entered in degrees and hundredths with decimal in column 40
GMT correction	44	Obtain from DOD Flight Information Publication Enroute Supplement
Daylight saving time correction	46	See note for GMT correction

Last card must be followed by a card with "QUIT" in columns 1-4.

Leap Year

Input "YES" or "NO" on a separate card beginning in column 1

Schedule Day

Input on a separate card the Julian date for which the schedule is being prepared beginning in column 1

Daylight Saving Time

Input "YES" or "NO" on a separate card beginning in column 1

Aircraft Availability Data

Item	Card Column	Note
Location	1-4	Enter ICAO identifier of aircraft present location
Termination	6-9	Enter ICAO identifier of airfield at which aircraft will terminate its itinerary
Number	10	Enter the number of aircraft with the same location and termination point (If this number exceeds 9, enter remainder on second card)

The last card must be followed by a card with "QUIT" in columns 1-4 and 6-9, and 0 in column 10.

Travel Request Data

Item	Card Column	Note
Origin	1-4	Enter ICAO identifier of origin
Destination	6-9	Enter ICAO identifier of destination
NET date	11-13	Julian date
NET time	15-18	24-hour clock

Item	Card Column	Note
NLT date	20-22	Julian date
NLT time	24-27	24-hour clock
Passenger load	29	
Request priority	31-32	Entry must be right-justified
DV code	34	For personnel with no DV status, enter "9" in column 34
Passenger name	36-50	Enter last name first
Passenger rank	51-54	Enter rank and branch of service in manner currently in use at MAC/DOOF

Last card must be followed by card with "QUIT" in columns 1-4

Manual Intervention

To insure that a priority 1-3 request is selected as the primary request ahead of all others (except USAF 0-10s), enter a card with the information below. If manual intervention is desired for more than one request, enter these cards in the preferred sequence of selection.

Item	Card Column	Note
Name	1-10	Enter first 10 characters of passenger's name exactly as they appear on the travel request card
Aircraft Location	11-14	Enter ICAO identifier of present location of aircraft designated to support request (optional-if aircraft is not designated or designated aircraft is not available, algorithm will select aircraft to support request)
Aircraft Termination point	15-18	Enter ICAO identifier of termination point of aircraft selected in column 11-14 (Leave blank if columns 11-14 are blank)

Manual Manipulation

Manual Splitting. A priority 1-3 travel request may be manually split into two travel requests to allow it to be supported by an aircraft supporting another priority 1-3 request going in the same general direction at a compatible time. An example is for a request from A to C to be split into a request from A to B and from B to C. The original request is discarded and two new requests are prepared using the format above. The only difference is that the request from B to C must contain "QQQQQ" in columns 46-50 (if the scheduler wants to guarantee that the same aircraft carries the passenger from A-B and B-C). If "QQQQQ" is omitted the result is a possible manual interplane, and whether the passenger will remain on the aircraft to termination depends on crew day and secondary request selection criteria.

Request Removal. To avoid manually splitting a request, the scheduler may obtain similar results by simply removing the request from the file and manually scheduling it.

Changing NLT Times. To advance a priority 1-3 request in the primary request selection sequence, the NLT time may be made earlier prior to running the model. This will not affect the departure time, since that is based on NET time. (When changing the NLT time, insure that there is more than one minute difference between the NLT and NET times).

Section II. Sample Output

This section consists of a series of figures displaying samples of model output. Most figures are self-explanatory. Any which require explanation are annotated, and the explanation follows on the next page.

AIRPORTS IN DATA BASE

NAME	ICAO IDENTIFIER	NORTH LATITUDE DEG MIN	WEST LONGITUDE DEG MIN	GMT TIME CORRECTIONS STD	GMT TIME CORRECTIONS DST
ALLEN C. THOMPSON FIELD	KJAN	32.19	90.05	5	5
ALTUS AFB	KLTS	34.40	99.16	6	5
ANDREWS AFB	KADW	38.49	76.52	5	4
BALTIMORE-WASHINGTON INTL	KBWI	39.07	76.40	5	4
BARKSDALE AFB	KBAD	32.30	93.40	6	5
BEALE AFB	KBAB	39.08	121.26	8	7
BERGSTROM AFB	KBSM	30.13	97.40	6	5
BLYTHEVILLE AFB	KBYH	35.58	89.57	6	5
BROOME CO	KBGM	42.13	75.59	5	4
BUCKLEY ANGB	KBKF	39.43	104.45	7	6
CANNON AFB	KCVS	34.23	103.19	7	6
CARSWELL AFB	KFWH	32.47	97.26	6	5
CASTLE AFB	KMER	37.23	120.34	8	7
CHARLESTON AFB	KCHS	32.54	80.02	5	4
CHICAGO-O'HARE IAP	KORD	41.59	87.54	6	5
COLUMBUS AFB	KCBM	33.39	88.27	6	5
CORPUS CHRISTI IAP	KCRP	27.46	97.30	6	5
DAVIS-MONTHAN AFB	KDMA	32.10	110.53	7	7
DOBBINS AFB	KMGE	33.55	84.31	5	4
DOVER AFB	KDOV	39.08	75.28	5	4
DULUTH IAP	KDLH	46.50	92.11	6	5
DYESS AFB	KDYS	32.26	99.51	6	5
EDWARDS AFB	KEDW	34.54	117.52	8	7
EGLIN AFB	KVPS	30.29	86.32	6	5
ELLINGTON AFB	KEFD	29.37	95.10	6	5
ELLSWORTH AFB	KRCA	44.09	103.06	7	6
ENGLAND AFB	KAEX	31.20	92.33	6	5

Figure B-1. Airports in Data Base

LEAP YEAR	DAY	DAYLIGHT SAVING TIME	T-39S AVAILABLE
NO	5	NO	13

SABRELINERS AVAILABLE FOR OPERATIONAL SUPPORT AIRLIFT

LOCATION	TERM POINT	NUMBER
KADW	KADW	2
KVPS	KVPS	1
KHIF	KOFF	1
KLFI	KLFI	2
KLAX	KFFO	1
KMCC	KMCC	2
KNUQ	KBAD	1
KOFF	KOFF	1
KCOS	KCOS	1
KSSC	KSSC	1

Figure B-2. Schedule Day and Aircraft Availability Data

PRIORITY 1-3 TRAVEL REQUESTS WITH ORIGIN AND DESTINATION IN BASE FILE

ORGIN ICAO-ID	DEST ICAO-ID	NET DAY	NET TIME	NLT DAY	NLT TIME	NO. PAX	PAX PRIORITY	DV CODE	NAME-FIRST PASSENGER	RANK/ SERVICE
KADW	KCOF	6	1400	6	1630	1	3	5	MEADE, H	08/A
KHIF	KADW	6	2100	6	2101	3	2	2	HATCH, O	CC/S
KHIF	KFFO	6	2300	7	600	5	3	5	ABRAHAMSON, J	08/A
KLAX	KADW	6	1700	6	1701	2	3	4	STAFFORD, T	09/A
KMGE	KADW	6	1400	6	1700	4	3	5	DILLON, E	08/A
KNUQ	KBLV	6	1800	6	2400	2	3	6	MACLAREN, W	07/A

PRIORITY 4-12 TRAVEL REQUESTS WITH ORIGIN AND DESTINATION IN BASE FILE

ORGIN ICAO-ID	DEST ICAO-ID	NET DAY	NET TIME	NLT DAY	NLT TIME	NO. PAX	PAX PRIORITY	DV CODE	NAME-FIRST PASSENGER	RANK/ SERVICE
KADW	KCOS	6	2000	7	300	1	4	9	PFISTER, J	04/A
KADW	KCOS	6	1200	7	400	1	5	7	CORRIGAN, W	06/A
KADW	KCOS	6	1300	6	2300	1	11	9	ROTTER, J	CC/A
KADW	KOFF	6	2100	6	2300	1	6	9	MATHIS, W	04/A
KADW	KOFF	7	300	7	645	5	4	9	BRENNEN, H	04/A

TRAVEL REQUESTS WITH ORIGINS OR DESTINATIONS NOT IN BASE FILE
BASES MUST BE ADDED TO BASE FILE OR REQUESTS MUST BE REMOVED OR CHANGED

ORGIN ICAO-ID	DEST ICAO-ID	NET DAY	NET TIME	NLT DAY	NLT TIME	NO. PAX	PAX PRIORITY	DV CODE	NAME-FIRST PASSENGER	RANK/ SERVICE
KILG	KBLV	6	800	6	1300	4	7	9	BJERKEN, P	04/A
KMXF	KVNY	6	1300	6	2300	4	7	9	BABCOOCK, G	CC/A
KNUQ	CSLI	6	2200	6	2400	1	8	9	SHURM, J	CC/A
KLSV	KNKX	6	1400	7	700	1	8	9	CABELLO, J	CC/A

Figure B-3. Travel Request Data

PRIORITY 1-3 REQUESTS WILL NORMALLY BE SCHEDULED IN THIS ORDER.

KFFO	KADW	250	10.00	250	12.00	1	3	5	EWING R	CC/A
KADW	KCMH	250	11.00	250	12.09	4	3	4	POST, G	09/A
KFFO	KBOS	250	10.00	250	12.50	5	3	6	KOEPNICK, D	CC/A
KADW	KFFO	250	11.00	250	12.50	5	3	5	KULP, B	CC/A
KFFO	KADW	250	10.75	250	13.00	4	3	6	URBAN, L	CC/A
KBSM	KSSC	250	11.00	250	14.00	2	3	5	RYAN, M	08/A
KADW	KFFO	250	12.00	250	14.25	2	3	6	GOLDFARB, O	CC/A
KOFF	KBKF	250	13.00	250	14.50	5	3	5	LARSON, D	08/A
KRME	KFFO	250	11.50	250	14.75	2	3	6	BURGESS, J	CC/A
KCOS	KLUF	250	13.50	250	15.25	3	3	3	HILL, J	09/A

REQUESTS WERE SCHEDULED IN THIS ORDER AND LOADED ON INDICATED SARFELINERS

#T39	LOCATION	TERM PT	NAME	ORIGIN	DEST	PRI	DVC	RANK	SELECTED
39	KCOS	KCOS	HILL, J	KCOS	KLUF	3	3	08/A	SCHED MAN
38	KMCC	KADW	CREEDON, J	KLUF	KADW	3	6	07/A	SCHEDULER
37	KVBG	KADW	MACLAREN, W	KVBG	KBLV	3	5	07/A	SCHEDULER
36	KSSC	KSBD	CAPR, R	KADW	KLUF	3	5	08/A	SCHEDULER
35	KFFO	KHIF	MULLINS, J	KFFO	KHIF	3	5	08/A	SCHEDULER
34	KADW	KABQ	BAKER, M	KADW	KABQ	3	6	07/A	SCHEDULER
33	KABQ	KVPS	LEAF, H	KABQ	KVPS	3	5	08/A	SCHEDULER
32	KRND	KRND	MCIVER, R	KADW	KRND	3	5	07/A	SCHEDULER
31	KBLV	KBLV	TALLMAN, K	KCOS	KSKF	3	4	09/A	SCHEDULER
30	KSSC	KSSC	CURRY,W	KRND	KRIC	3	5	07/A	SCHEDULER
29	KOFF	KOFF	MASTERSON, W	KBAD	KBYH	3	6	07/A	SCHEDULER
28	KFFO	KFFO	EWING R	KFFO	KADW	3	5	CC/A	ALGORITHM

(See Explanation on Following Page)

Figure B-4. Normal and Actual Selection Sequence for Priority 1-3 Requests

The "SELECTED" column on the preceding page indicates the manner in which the request was selected as a primary request:

SCHED MAN	indicates that the request is from a USAF 0-10 and that the scheduler should manually schedule the mission for that aircraft
SCHEDULER	indicates that the request was directed to be scheduled at that point by manual intervention
ALGORITHM	indicates that the request was selected as the primary request by algorithm selection criteria

```
INFEASIBLE:    4 T-39S ARE NOT SCHEDULED     0 REQUESTS ARE NOT SATISFIED

AIRCRAFT SCHEDULE FOR JULIAN DAY   7

#################################################################################

ANDREWS AFB

                                                 PASSENGER LOAD DATA
DEPART TIME(Z) ARRIVE TIME(Z) #  RANKING PASSENGER   PRIORITY DESTINATION
---------------------------------------------------------------------------------
CREW DUTY START    GMT DAY  GMT TIME     LOCAL DAY  LOCAL TIME
                     7       1417            7          917

 KADW   1617   KWRB   1800
 KWRB   1900   KADW   2027
                              2  08/A BODYCOMBE, R       3        KADW
 KADW   2200   KMXF      3
                              1  09/A GREENLEAF, A       3        KMXF
 KMXF    103   KADW    243
     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     +UNSATISFIED REQUESTS THAT MAY BE COMPATIBLE WITH THIS ROUTING+
     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

---------------------------------------------------------------------------------
```

Infeasible is a label that indicates the end of the T-39 Scheduler routine

Figure B-5. Normal Schedule Printout

```
--------------------------------------------------------------------------------
     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     X NOTE: PASSENGER IS AN O-10.  MANUALLY SCHEDULE MISSION. X
     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CREW DUTY START   GMT DAY  GMT TIME    LOCAL DAY  LOCAL TIME
                    250      1130         250        530

 KCOS   1330   KLUF   1506
                                3  O0/A HILL, J             3       KLUF
 KLUF   2000   KCOS   2145
                                3  O0/A HILL, J             3       KCOS
--------------------------------------------------------------------------------
################################################################################

SHAW AFB

                                PASSENGER LOAD DATA
DEPART TIME(Z) ARRIVE TIME(Z) #  RANKING PASSENGER   PRIORITY DESTINATION
--------------------------------------------------------------------------------
CREW DUTY START   GMT DAY  GMT TIME    LOCAL DAY  LOCAL TIME
                    250      1252         250        852

 KSSC   1452   KADW   1600
 KADW   1700   KIAB   1953
                                1  O8/A CARR, R             3       KLUF
                                4  O7/A WILLARD, G          3       KLUF
 KIAB   2053   KLUF   2315
                                1  O8/A CARR, R             3       KLUF
                                4  O7/A WILLARD, G          3       KLUF
 KLUF     17   KSBD    115
                                4  O7/A WILLARD, GQQQQQ     3       KSBD
    +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        "QQQQQ" following passenger name indicates manually-split request
```

Figure B-6. Sample O-10 Mission and Sample Manually-Split Mission

```
-----------------------------------------------------------------------------------------
CREW DUTY START    GMT DAY  GMT TIME    LOCAL DAY  LOCAL TIME
                      250      1325          250        925

 KLFI    1525    KADW    1600
 KADW    1700    KSSC    1811
                                   1  08/A CARR, R    XXXXX    3          KBSM
 KSSC    1911    KBSM    2154
                                   1  08/A CARR, R    XXXXX    3          KBSM
                                   1  08/A RYAN, M             3          KBSM
 KBSM    2254    KLFI     155
     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     +UNSATISFIED REQUESTS THAT MAY BE COMPATIBLE WITH THIS ROUTING+
     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

 KLFI    1200    KSSC    2400      1  03/A SLADE, W           12          KSSC
 KLFI    1200    KBSM     500      1  05/A BOYCE, J           12          KBSM

 KLFI    1200    KBSM    1700      3  04/A KNEDLIK, D         12          KBSM
 KADW    1700    KBSM    2210      4  07/A WILLARD, GZZZZZ     3          KBSM
-----------------------------------------------------------------------------------------
```

"XXXXX" following passenger name indicates request was created by Interplane routine

"ZZZZZ" following passenger name indicates request that could be interplaned along the same route as the primary request

Figure B-7. Sample Interplane Mission

```
CREW DUTY START   GMT DAY  GMT TIME    LOCAL DAY  LOCAL TIME
                     7       1355           7         755

 KBSM   1555   KADW   1845
THE LEG BELOW REQUIRES MANUAL SELECTION OF 1 REFUELING BASE

 KADW   1945   KSBD    312
                                1  09/A LANE, H           3       KSBD
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     +UNSATISFIED REQUESTS THAT MAY BE COMPATIBLE WITH THIS ROUTING+
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

--------------------------------------------------------------------------------
```

Figure B-8.

Sample of Mission that requires Manual Selection of Refueling Base

```
--------------------------------------------------------------------------------------
CREW DUTY START    GMT DAY  GMT TIME    LOCAL DAY  LOCAL TIME
                      250      1800         250        1100

 KOFF    1800   KADW    2024
                                5  08/A O'MALLY, J         3          KADW
 KADW    2124   KCOF    2324
                                3  CC/A SINGIEVICH, W      3          KCOF
 KCOF      24   KOFF     331
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     +UNSATISFIED REQUESTS THAT MAY BE COMPATIBLE WITH THIS ROUTING+
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++



 KOFF    1615   KADW    2200    1  06/A LANCASTER, A      12          KADW
 KOFF    1200   KADW    2400    1  05/A GAUTHIER, R       12          KADW
 KADW    1800   KCOF    2100    1  04/A JERRICK, T        12          KCOF
 KADW    1145   KOFF    1730    3  06/A BENDORF, H         4          KOFF
 KADW    1845   KOFF     500    1  05/A GREENE, J         12          KOFF  ◄
 KADW    2130   KOFF    1700    1  04/A SCHWALM, A        12          KOFF  ◄
 KADW    1845   KOFF    2200    1  CC/A RICHARDS, G       12          KOFF
--------------------------------------------------------------------------------------
```

Figure B-9.

Sample of mission with Additional Requests that Can Be Loaded Directly

UNSUPPORTED PRIORITY 1-3 TRAVEL REQUESTS

ORGIN ICAO-ID	DEST ICAO-ID	NET DAY	NET TIME	NLT DAY	NLT TIME	NO. PAX	PAX PRIORITY	DV CODE	NAME-FIRST PASSENGER	RANK/

UNSUPPORTED PRIORITY 4-12 TRAVEL REQUESTS

ORGIN ICAO-ID	DEST ICAO-ID	NET DAY	NET TIME	NLT DAY	NLT TIME	NO. PAX	PAX PRIORITY	DV CODE	NAME-FIRST PASSENGER	RANK/ SERVICE
KADW	KCOS	6	2000	7	300	1	4	9	PFISTER, J	04/A
KADW	KCOS	6	1200	7	400	1	5	7	CORRIGAN, W	06/A
KADW	KCOS	6	1300	6	2300	1	11	9	ROTTIER, J	CC/A
KDYS	KSBD	6	1300	7	800	1	5	9	KIRKPATRICK, C	04/A
KFFO	KCOF	6	2000	6	2200	1	9	9	DANIELS, T	04/A
KFFO	KDMA	6	1800	7	300	4	6	7	BATTAGLIA, J	06/A
KFFO	KWRB	6	1300	6	2200	1	12	9	SMITH, O	CC/A
KHIF	KMCC	7	130	7	400	1	6	9	SHEFFIELD, R	03/A
KLFI	KBSM	6	1200	6	2200	1	12	9	CARNE, R	05/A
KLSV	KOFF	6	1600	6	2400	1	12	9	WALKER, E	CC/A

TRAVEL REQUESTS THAT MAY STILL BE SUPPORTED IN THE FUTURE

ORGIN ICAO-ID	DEST ICAO-ID	NET DAY	NET TIME	NLT DAY	NLT TIME	NO. PAX	PAX PRIORITY	DV CODE	NAME-FIRST PASSENGER	RANK/ SERVICE
KABQ	KOFF	250	2300	252	300	1	6	9	REESE, L	04/A
KADW	KFFO	250	1900	251	1300	1	8	9	STALLKAMP, R	05/A
KEDW	KABQ	250	200	251	1800	1	8	9	HEGSTROM, R	04/A
KVPS	KFFO	250	2300	251	1800	1	9	9	THOMPSON, R	04/A
KVPS	KFFO	250	2300	251	2000	2	9	9	BUICK, J	05/A
KLAX	KABQ	250	2000	251	2000	1	9	9	MALONEY, W	CC/A

Figure B-10. Unsupported Request Data

Section III. A Technique for Using the Model

This section outlines a technique for using the model in its existing form. This technique can be streamlined if the recommendations of Chapter V are followed and the model is adapted for interactive use.

To use the model to prepare an initial schedule, the scheduler may proceed as follows:

a. Prepare the data as described in this appendix.

b. Enter only the priority 1-3 requests. Run the model with no manual interventions. (Using all the requests would not significantly alter the final schedule, but would detract from the scheduler's ability to quickly identify the most desirable manual interventions.)

c. Evaluate the output. If all priority 1-3 requests are not scheduled, look for requests that require long travel times. If possible, assign these requests to aircraft which will terminate their itineraries at points other than their initial locations. Be sure to consider the earliest times of availability for aircraft that have remained away from their home stations. Also consider the possibility of designating additional aircraft to RON away from their home stations.

d. Make repeated runs of the model. Use manual intervention to evaluate alternative scheduling strategies until all priority 1-3 requests can be supported. If attempts at manual intervention do not produce a plan that

will support all priority 1-3 requests, consider what changes to travel times will allow all requests to be supported. If some minor changes to travel times will allow all requests to be supported, investigate with the affected requesters the possibility of changing their planned travel times.

e. Iterate the process until arriving at a plan that will support as many priority 1-3 requests as possible. Add the priority 4-12 requests and run the model.

f. Evaluate the resulting schedule and determine if assigning particular priority 4-12 requests to specific aircraft would enable more passengers to be carried without affecting support for priority 1-3 requests.

g. Manually assign additional request to specific itineraries that have enough empty seats. Possible ways to load more passengers are:

1. Check the list under each itinerary of "Unsatisfied Requests That May Be Compatible With This Routing." Some requests can be loaded directly. Some may be supported if the passenger load is reduced. Some may be loaded if the time window between the NET and NLT times is expanded. A rule of thumb is to load any request that needs to be changed by no more than 15 minutes. Investigate larger changes with the requesters.

2. Determine how much crew duty time each mission requires. Missions with several hours of unused crew duty time may be able to support additional requests by making an earlier takeoff.

AD-A069 744 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 15/7
A COMPUTERIZED TECHNIQUE FOR SCHEDULING MILITARY AIRLIFT COMMAN--ETC(U)
MAR 79 G P MILNE, R K COFFEY

UNCLASSIFIED AFIT/GST/SM/79M-3 NL

2 OF 2
AD
A069744

END
DATE
FILMED
7-79
DDC

3. Check for a long ground time between legs. Some of this time can possibly be used to support requests which originate and terminate along the route of the later leg.

4. Consider reducing scheduled ground times when refueling is not required. This extra 30 minutes may often be used to support an additional request.

5. Check the final schedule for the status of airfields used. Secure prior permission to transit airfields which require it. Make any adjustments necessary. The initial schedule is now complete.

APPENDIX C

Use of the Model in Day 250 Schedule Preparation

The schedule for day 250 was prepared by generally following the steps for model use outlined in the user's guide. For this day, there were 39 Sabreliners available at 16 bases. Seven of these aircraft were identified to terminate their itineraries at points other than their initial locations. There were 54 priority 3 and 242 priority 4-12 requests to be scheduled. Two of the priority 3 requests were submitted by a USAF 0-10.

These steps were used in producing the schedule:

a. The model was run with only the priority 3 requests and with no manual intervention. Five requests were not supported, and one aircraft was not scheduled.

b. Six requests requiring long travel times were manually assigned to specific aircraft which had been identified to terminate their itineraries at points other than their initial locations. The model was run a second time. This time, three requests were not supported and one aircraft was not scheduled.

c. Examination of the results of the second run indicated that a request from Andrews AFB, MD to Norton AFB, CA could be loaded on the same Sabreliner with a request from Andrews to Luke AFB, AZ. This aircraft was terminating its itinerary at Norton. In accordance with the user's

guide, the request from Andrews to Norton was manually split into one request from Andrews to Luke and another request from Luke to Norton. This same result could have been achieved by removing the request and adding it to the mission manually.

d. The model was run a third time. One aircraft from Langley AFB, VA was not scheduled, and General A's request to travel from Randolph AFB, TX to Byrd International Airport was not supported. Examining the output revealed that one of the last requests being scheduled by the algorithm, General B's request to travel from Peterson AFB, CO to Randolph was being interplaned at Vance AFB, OK. This situation developed because the only unscheduled Sabreliners remaining at that time were at McClellan AFB, CA and Langley. This request could easily be handled by a single aircraft from several closer operating locations, so manual instructions were entered to select General B's request as the ninth primary request instead of the 36th. No particular aircraft was specified because there was no reason to override the algorithm selection rules.

e. The model was run for the fourth time. Two aircraft were not used, but General A's request was still not supported. Manual instructions were entered to select General A's request as the primary request immediately after General B's. The model was run again, and all priority 3 requests were supported. Additionally, only 36 of the 39 Sabreliners available were required to support all the

priority 3 requests.

f. The priority 4-12 requests were added, and the model was run for the sixth time. A total of 274 passengers were supported.

g. Following the steps outlined in the user's guide for manual passenger loading, 16 more passengers were added. The initial schedule was now complete and 290 passengers were supported.

APPENDIX D

Program Listing

```
PREAMBLE

''THE PREAMBLE SPECIFIES THE PHYSICAL ELEMENTS OF THE SYSTEM AND
''THE RELATIONSHIPS AMONG THESE ELEMENTS
TEMPORARY ENTITIES
EVERY BASE HAS
     A NAME1,A NAME2,A NAME3,
     AN ICAO.IDENTIFIER,
     A NORTH.LATITUDE,
     A WEST.LONGITUDE,
     A GMT.CORRECTION,
     A DST.GMT.CORRECTION,
     A LCL.TIME.CHANGE,
     BELONGS TO A BASE.FILE,
     AND MAY OWN A DET
''A DET IS A MAC DETACHMENT
EVERY RF.BASE HAS
     AN RF.ICAO,
     AN IN.TIME,
     AN IN.FUEL,
     AN OUT.TIME,
     AN OUT.FUEL,
     A TOT.TIME,
     A PAX.VALUE,
     AND BELONGS TO A REF.BASE.FILE
EVERY SABRELINER HAS
     A HOME.STATION,
     A CREW.DUTY.START,
     A MAX.DUTY.DAY,
     A DUTY.DAY,
     A SEATS.AVAILABLE,
     BELONGS TO A DET,
     AND OWNS AN ITINERARY
''HOME.STATION NORMALLY REFERS TO THE SABRELINER'S BASE OF ASSIGNMENT
''HOWEVER, HOME.STATION MAY BE USED TO IDENTIFY ANY BASE AT WHICH THE
```

```
''SCHEDULER WANTS THE SABRELINER TO REMAIN OVERNIGHT
EVERY LEG HAS
     AN L.ORIGIN,
     AN L.DESTINATION,
     A DEPARTURE.TIME,
     AN ENROUTE.TIME,
     AN ARRIVAL.TIME,
     A FUEL.CONSUMED,
     BELONGS TO AN ITINERARY,
     AND MAY OWN SOME SATISFIED.REQUESTS
EVERY TRAVEL.REQUEST HAS
     AN R.ORIGIN,
     AN R.DESTINATION,
     AN NET.DATE,
     AN NET.TIME,
     AN NLT.DATE,
     AN NLT.TIME,
     A PAX.LOAD,
     A PAX.PRIORITY,
     A DV.CODE,
     A PAX.1NAME,
     A PAX.2NAME,
     A PAX.RANK,
     AND MAY BELONG TO THE SATISFIED.REQUESTS,
          THE UNSATISFIED.REQUESTS,
          THE PRT.UNSAT.REQ,
          AND THE PENDING.REQUESTS
DEFINE BASE.FILE AS A SET RANKED BY LOW NAME1
DEFINE REF.BASE.FILE AS A SET RANKED BY HIGH PAX.VALUE,
     THEN BY LOW TOT.TIME
DEFINE DET AS A SET RANKED BY LOW CREW.DUTY.START
DEFINE ITINERARY AS A SET RANKED BY LOW DEPARTURE.TIME
DEFINE SATISFIED.REQUESTS,
     UNSATISFIED.REQUESTS,
     AND PENDING.REQUESTS AS SETS RANKED BY LOW PAX.PRIORITY,
          THEN BY LOW NLT.DATE,
```

```
          THEN BY LOW NLT.TIME,
          THEN BY LOW PAX.1NAME
DEFINE PRT.UNSAT.REQ AS A SET RANKED BY  LOW R.ORIGIN,
     THEN BY LOW R.DESTINATION,
     THEN BY LOW PAX.PRIORITY,
     THEN BY LOW NET.DATE,
     THEN BY LOW NET.TIME
THE SYSTEM OWNS THE BASE.FILE,
     THE REF.BASE.FILE,
     THE UNSATISFIED.REQUESTS,
     THE PRT.UNSAT.REQ,
     AND THE PENDING.REQUESTS
DEFINE
     NAME1,NAME2,NAME3,
     ICAO.IDENTIFIER,
     RF.ICAO,
     DAYLIGHT.SAVING.TIME,
     HOME.STATION,
     L.ORIGIN,R.ORIGIN,
     L.DESTINATION,R.DESTINATION,
     PAX.1NAME,
     PAX.2NAME,
     PAX.RANK
     AS ALPHA VARIABLES
DEFINE
     GMT.CORRECTION,
     DST.GMT.CORRECTION,
     LCL.TIME.CHANGE,
     NET.DATE,
     NLT.DATE,
     PAX.LOAD,
     PAX.PRIORITY,
     SEATS.AVAILABLE,
     DV.CODE,
     IPRQPT,
     DAY,
```

```
    YEAR.DAYS,
    T39S.AVAILABLE
    AS INTEGER VARIABLES
DEFINE ICAO TO MEAN ICAO.IDENTIFIER
DEFINE LAT TO MEAN NORTH.LATITUDE
DEFINE LONG TO MEAN WEST.LONGITUDE
DEFINE GMT TO MEAN GMT.CORRECTION
DEFINE DST TO MEAN DST.GMT.CORRECTION
DEFINE REQ TO MEAN TRAVEL.REQUEST
END ''OF PREAMBLE
```

```
MAIN

CALL READ.DATA
CALL T39.SCHEDULER
CALL PRINT.SCHEDULE
STOP
END ''OF MAIN
```

```
ROUTINE TO READ.DATA


DEFINE
     I,
     BASE1,
     NO.T39S
     AS INTEGER VARIABLES
DEFINE
     LEAP.YEAR,
     LOCATION,
     HOMEBASE
     AS ALPHA VARIABLES
''INPUT AIRPORT DATA
''LAT AND LONG MUST BE ENTERED IN DEGREES AND HUNDREDTHS OF DEGREES
''FOR EXAMPLE, 30 DEGREES AND 15 MINUTES MUST BE ENTERED AS 30.25
''LAST BASE CARD MUST BE FOLLOWED BY A CARD WITH "QUIT" IN COLUMNS 1-4
'BASE.DATA'
CREATE A BASE
START NEW CARD
READ NAME1(BASE),NAME2(BASE),NAME3(BASE),
     ICAO(BASE),
     LAT(BASE),
     LONG(BASE),
     GMT(BASE),
     AND DST(BASE)
     AS 2 A 10,A 5,A 4,D(6,2),D(7,2),2 I 2
IF NAME1(BASE) NE "QUIT"
     FILE BASE IN BASE.FILE
     GO TO BASE.DATA
ELSE
     DESTROY BASE
PRINT 4 LINES THUS
AIRPORTS IN DATA BASE
                                      NORTH      WEST       GMT TIME
                             ICAO    LATITUDE  LONGITUDE  CORRECTIONS
```

```
          NAME           IDENTIFIER  DEG  MIN  DEG   MIN   STD   DST
FOR EACH BASE IN BASE.FILE
DO
     LET X=TRUNC.F(LAT(BASE))+(LAT(BASE)-TRUNC.F(LAT(BASE)))*.6
     LET Y=TRUNC.F(LONG(BASE))+(LONG(BASE)-TRUNC.F(LONG(BASE)))*.6
     WRITE NAME1(BASE),NAME2(BASE),NAME3(BASE),
          ICAO(BASE),
          X,Y,
          GMT(BASE),
          AND DST(BASE)
          AS 2 A 10,A 8,A 4,2 D(11,2),I 7,I 6,/
LOOP
''INPUT "YES" OR "NO" FOR LEAP YEAR
READ LEAP.YEAR
IF LEAP.YEAR EQ "YES"
     LET YEAR.DAYS=366
ELSE
     LET YEAR.DAYS=365
ALWAYS
''INPUT JULIAN DAY FOR WHICH SCHEDULE IS TO BE PREPARED
READ DAY
''INPUT "YES" OR "NO" FOR DAYLIGHT SAVING TIME
READ DAYLIGHT.SAVING.TIME
IF DAYLIGHT.SAVING.TIME EQ "YES"
     FOR EACH BASE IN BASE.FILE
          LET LCL.TIME.CHANGE(BASE)=DST(BASE)
ELSE
     FOR EACH BASE IN BASE.FILE
          LET LCL.TIME.CHANGE(BASE)=GMT(BASE)
ALWAYS
''INPUT AIRCRAFT LOCATIONS, TERMINATION BASES, AND NUMBERS AVAILABLE
''LAST ENTRY MUST BE FOLLOWED BY "QUIT QUIT 0"
'T39.DATA'
START NEW CARD
READ LOCATION,
     HOMEBASE,
```

```
        NO.T39S
        AS A 5,A 5,I 1
   LET T39S.AVAILABLE=T39S.AVAILABLE+NO.T39S
   FOR EACH BASE IN BASE.FILE,WITH ICAO(BASE) EQ LOCATION,
        FIND THE FIRST CASE
        IF FOUND,
             IF LOCATION NE HOMEBASE
                  CREATE A BASE CALLED BASE1
                  LET NAME1(BASE1)=NAME1(BASE)
                  LET NAME2(BASE1)=NAME2(BASE)
                  LET NAME3(BASE1)=NAME3(BASE)
                  LET ICAO(BASE1)=ICAO(BASE)
                  LET LAT(BASE1)=LAT(BASE)
                            LET LONG(BASE1)=LONG(BASE)
                  LET GMT(BASE1)=GMT(BASE)
                            LET DST(BASE1)=DST(BASE)
                  LET LCL.TIME.CHANGE(BASE1)=LCL.TIME.CHANGE(BASE)
                  FILE BASE1 IN BASE.FILE
                  FOR I=1 TO NO.T39S
                  DO
                       CREATE A SABRELINER
                       LET HOME.STATION(SABRELINER)=HOMEBASE
                       LET SEATS.AVAILABLE(SABRELINER)=5
                       FILE SABRELINER IN DET(BASE1)
                  LOOP
                  GO TO T39.DATA
             ELSE
             FOR I=1 TO NO.T39S
             DO
                  CREATE A SABRELINER
                  LET HOME.STATION(SABRELINER)=HOMEBASE
                  LET SEATS.AVAILABLE(SABRELINER)=5
                  FILE SABRELINER IN DET(BASE)
             LOOP
             GO TO T39.DATA
        ELSE
```

```
PRINT 4 LINES WITH LEAP.YEAR,DAY,DAYLIGHT.SAVING.TIME,T39S.AVAILABLE
     THUS

LEAP YEAR  DAY  DAYLIGHT SAVING TIME  T-39S AVAILABLE
   ***     ***            ***               ***

PRINT 4 LINES THUS

SABRELINERS AVAILABLE FOR OPERATIONAL SUPPORT AIRLIFT

LOCATION   TERM POINT   NUMBER
FOR EACH BASE IN BASE.FILE,WITH N.DET(BASE) GT 0,
DO
     PRINT 1 LINE WITH ICAO(BASE),
          HOME.STATION(F.DET(BASE)),
          N.DET(BASE)
          THUS
  ****        ****        **
     LOOP
''INPUT TRAVEL REQUESTS ON INDIVIDUAL CARDS
''ORIGINS AND DESTINATIONS ARE ICAO IDENTIFIERS
''DATES ARE JULIAN DAYS
''TIMES ARE GMT BASED ON A 24-HOUR CLOCK
''LAST REQUEST CARD MUST BE FOLLOWED BY A CARD WITH "QUIT" IN
''COLUMNS 1-4
'REQUEST.DATA'
CREATE A TRAVEL.REQUEST
START NEW CARD
READ
     R.ORIGIN(TRAVEL.REQUEST),
     R.DESTINATION(TRAVEL.REQUEST),
     NET.DATE(TRAVEL.REQUEST),
     NET.TIME(TRAVEL.REQUEST),
     NLT.DATE(TRAVEL.REQUEST),
     NLT.TIME(TRAVEL.REQUEST),
     PAX.LOAD(TRAVEL.REQUEST),
```

```
      PAX.PRIORITY(TRAVEL.REQUEST),
      DV.CODE(TRAVEL.REQUEST),
      PAX.1NAME(TRAVEL.REQUEST),
      PAX.2NAME(TRAVEL.REQUEST),
      PAX.RANK(TRAVEL.REQUEST)
      AS A 5,A 4,I 4,I 5,I 4,I 5,I 2,I 3,I 2,S 1,A 10,A 5,A 4
IF R.ORIGIN(TRAVEL.REQUEST) NE "QUIT"
      FILE TRAVEL.REQUEST IN UNSATISFIED.REQUESTS
      GO TO REQUEST.DATA
ELSE
''IDENTIFY TRAVEL REQUESTS THAT CANNOT BE CONSIDERED
FOR EACH REQ IN UNSATISFIED.REQUESTS
DO
      FOR EACH BASE IN BASE.FILE,WITH ICAO(BASE) EQ R.ORIGIN(REQ),
            FIND THE FIRST CASE
            IF NONE
                  REMOVE REQ FROM UNSATISFIED.REQUESTS
                  FILE REQ IN PENDING.REQUESTS
            ALWAYS
LOOP
FOR EACH REQ IN UNSATISFIED.REQUESTS
DO
      FOR EACH BASE IN BASE.FILE,WITH ICAO(BASE) EQ R.DESTINATION(REQ),
            FIND THE FIRST CASE
            IF NONE
                  REMOVE REQ FROM UNSATISFIED.REQUESTS
                  FILE REQ IN PENDING.REQUESTS
            ALWAYS
LOOP
''PRINT TRAVEL REQUESTS TO BE CONSIDERED
PRINT 5 LINES THUS

PRIORITY 1-3 TRAVEL REQUESTS WITH ORIGIN AND DESTINATION IN BASE FILE

ORGIN   DEST    NET NET  NLT NLT  NO.   PAX     DV  NAME-FIRST          RANK/
ICAO-ID ICAO-ID DAY TIME DAY TIME PAX PRIORITY CODE PASSENGER          SERVICE
```

```
FOR EACH REQ IN UNSATISFIED.REQUESTS
     DO
          REMOVE REQ FROM UNSATISFIED.REQUESTS
          FILE REQ IN PRT.UNSAT.REQ
     LOOP
FOR EACH TRAVEL.REQUEST IN PRT.UNSAT.REQ WITH PAX.PRIORITY(REQ) LE 3
DO
WRITE
     R.ORIGIN(TRAVEL.REQUEST),
     R.DESTINATION(TRAVEL.REQUEST),
     NET.DATE(TRAVEL.REQUEST),
     NET.TIME(TRAVEL.REQUEST),
     NLT.DATE(TRAVEL.REQUEST),
     NLT.TIME(TRAVEL.REQUEST),
     PAX.LOAD(TRAVEL.REQUEST),
     PAX.PRIORITY(TRAVEL.REQUEST),
     DV.CODE(TRAVEL.REQUEST),
     PAX.1NAME(TRAVEL.REQUEST),
     PAX.2NAME(TRAVEL.REQUEST),
     PAX.RANK(TRAVEL.REQUEST)
     AS S 1,A 8,A 7,I 3,I 5,I 4,I 5,I 4,I 5,I 7,S 3,A 10,A 7,A 4,/
REMOVE TRAVEL.REQUEST FROM PRT.UNSAT.REQ
FILE TRAVEL.REQUEST IN UNSATISFIED.REQUESTS
LOOP
PRINT 5 LINES THUS

PRIORITY 4-12 TRAVEL REQUESTS WITH ORIGIN AND DESTINATION IN BASE FILE

ORGIN   DEST    NET NET  NLT NLT  NO.   PAX      DV  NAME-FIRST         RANK/
ICAO-ID ICAO-ID DAY TIME DAY TIME PAX PRIORITY CODE PASSENGER         SERVICE
FOR EACH TRAVEL.REQUEST IN PRT.UNSAT.REQ
DO
WRITE
     R.ORIGIN(TRAVEL.REQUEST),
     R.DESTINATION(TRAVEL.REQUEST),
     NET.DATE(TRAVEL.REQUEST),
```

```
    NET.TIME(TRAVEL.REQUEST),
    NLT.DATE(TRAVEL.REQUEST),
    NLT.TIME(TRAVEL.REQUEST),
    PAX.LOAD(TRAVEL.REQUEST),
    PAX.PRIORITY(TRAVEL.REQUEST),
    DV.CODE(TRAVEL.REQUEST),
    PAX.1NAME(TRAVEL.REQUEST),
    PAX.2NAME(TRAVEL.REQUEST),
    PAX.RANK(TRAVEL.REQUEST)
    AS S 1,A 8,A 7,I 3,I 5,I 4,I 5,I 4,I 5,I 7,S 3,A 10,A 7,A 4,/
REMOVE TRAVEL.REQUEST FROM PRT.UNSAT.REQ
FILE TRAVEL.REQUEST IN UNSATISFIED.REQUESTS
LOOP
''PRINT TRAVEL REQUESTS THAT CANNOT BE CONSIDERED
IF N.PENDING.REQUESTS GT 0
    PRINT 6 LINES THUS

TRAVEL REQUESTS WITH ORIGINS OR DESTINATIONS NOT IN BASE FILE
BASES MUST BE ADDED TO BASE FILE OR REQUESTS MUST BE REMOVED OR CHANGED

ORGIN   DEST    NET NET  NLT NLT  NO.   PAX     DV  NAME-FIRST        RANK/
ICAO-ID ICAO-ID DAY TIME DAY TIME PAX PRIORITY CODE PASSENGER        SERVICE
    FOR EACH TRAVEL.REQUEST IN PENDING.REQUESTS
    DO
        WRITE
            R.ORIGIN(TRAVEL.REQUEST),
            R.DESTINATION(TRAVEL.REQUEST),
            NET.DATE(TRAVEL.REQUEST),
            NET.TIME(TRAVEL.REQUEST),
            NLT.DATE(TRAVEL.REQUEST),
            NLT.TIME(TRAVEL.REQUEST),
            PAX.LOAD(TRAVEL.REQUEST),
            PAX.PRIORITY(TRAVEL.REQUEST),
            DV.CODE(TRAVEL.REQUEST),
            PAX.1NAME(TRAVEL.REQUEST),
            PAX.2NAME(TRAVEL.REQUEST),
```

```
                    PAX.RANK(TRAVEL.REQUEST)
                    AS S 1,A 8,A 7,I 3,I 5,I 4,I 5,I 4,I 5,I 7,S 3,A 10,
                         A 7,A 4,/
               REMOVE TRAVEL.REQUEST FROM PENDING.REQUESTS
               DESTROY TRAVEL.REQUEST
          LOOP
     ALWAYS
     FOR EACH REQ IN UNSATISFIED.REQUESTS
     DO
     ''ALIGN PASSENGER LOADS WITH PROJECTED AIRCRAFT PASSENGER CAPACITY
          IF PAX.LOAD(REQ) GT 5
               LET PAX.LOAD(REQ)=5
          ALWAYS
          IF NET.DATE(REQ) LT DAY AND DAY-NET.DATE(REQ) LE 15
               LET NET.DATE(REQ)=DAY
               LET NET.TIME(REQ)=0000
          ALWAYS
          IF NET.DATE(REQ) LT DAY AND DAY-NET.DATE(REQ) GT 15
               LET NET.DATE(REQ)=NET.DATE(REQ)+YEAR.DAYS
          ALWAYS
          IF NET.DATE(REQ) GT DAY AND NET.DATE(REQ)-DAY LE 15
               LET NET.TIME(REQ)=NET.TIME(REQ)+(NET.DATE(REQ)-DAY)*2400
          ALWAYS
          IF NET.DATE(REQ) GT DAY AND NET.DATE(REQ)-DAY GT 15
               LET NET.DATE(REQ)=DAY
               LET NET.TIME(REQ)=0000
          ALWAYS
          IF NLT.DATE(REQ) LT DAY
               LET NLT.DATE(REQ)=NLT.DATE(REQ)+YEAR.DAYS
          ALWAYS
          IF NLT.DATE(REQ) GT DAY
               LET NLT.TIME(REQ)=NLT.TIME(REQ)+(NLT.DATE(REQ)-DAY)*2400
          ALWAYS
     ''THIS SECTION DEALS WITH A CONVENTION BY WHICH HIGH PRIORITY TRAVELERS
     ''INDICATE AN INFLEXIBLE DEPARTURE OR ARRIVAL TIME
          IF PAX.PRIORITY(REQ) LE 3 AND ((NLT.TIME(REQ)-NET.TIME(REQ) EQ 1.)
```

```
        OR (TRUNC.F(NLT.TIME(REQ)/100)*100 EQ NLT.TIME(REQ)
        AND NLT.TIME(REQ)-NET.TIME(REQ) EQ 41.))
        CALL LEG.DATA GIVEN R.ORIGIN(REQ),R.DESTINATION(REQ)
             YIELDING X,I,TIME
        LET TIME=TIME+I*.25
        LET X=NET.TIME(REQ)/5
        LET Y=TRUNC.F(X)
        IF (X-Y) EQ 0.
             LET X=NLT.TIME(REQ)/100
             LET Y=TRUNC.F(X)
             LET NLT.TIME(REQ)=Y+(X-Y)*10/6+TIME
             LET X=TRUNC.F(NLT.TIME(REQ))
             LET Y=NLT.TIME(REQ)-X
             LET NLT.TIME(REQ)=(X+Y*6/10)*100
        ELSE
             LET X=NET.TIME(REQ)/100
             LET Y=TRUNC.F(X)
             LET NET.TIME(REQ)=Y+(X-Y)*10/6-TIME
             LET X=TRUNC.F(NET.TIME(REQ))
             LET Y=NET.TIME(REQ)-X
             LET NET.TIME(REQ)=(X+Y*6/10)*100
        ALWAYS
   ALWAYS
''CONVERT TIMES TO HOURS AND HUNDREDTHS
   LET X=NET.TIME(REQ)/100
   LET Y=TRUNC.F(X)
   LET NET.TIME(REQ)=Y+(X-Y)*10/6
   LET X=NLT.TIME(REQ)/100
   LET Y=TRUNC.F(X)
   LET NLT.TIME(REQ)=Y+(X-Y)*10/6
LOOP
''INSURE REQUESTS ARE RANKED  CORRECTLY AFTER BEING ADJUSTED FOR HARD
''TAKEOFF OR ARRIVAL TIMES.
FOR EACH REQ IN UNSATISFIED.REQUESTS
   DO
   REMOVE REQ FROM UNSATISFIED.REQUESTS
```

```
     FILE REQ IN PENDING.REQUESTS
     LOOP
FOR EACH REQ IN PENDING.REQUESTS
     DO
     REMOVE REQ FROM PENDING.REQUESTS
     FILE REQ IN UNSATISFIED.REQUESTS
     LOOP
''PRINT PRI 1-3 REQUESTS RANKED BY LOW NLT.TIME.  THIS IS THE NORMAL
''ORDER IN WHICH THEY ARE SCHEDULED.
SKIP 1 LINE
PRINT 1 LINE THUS
PRIORITY 1-3 REQUESTS WILL NORMALLY BE SCHEDULED IN THIS ORDER.
SKIP 1 LINE
FOR EACH TRAVEL.REQUEST IN UNSATISFIED.REQUESTS WITH
     PAX.PRIORITY(TRAVEL.REQUEST) LE 3
WRITE
     R.ORIGIN(TRAVEL.REQUEST),
     R.DESTINATION(TRAVEL.REQUEST),
     DAY,
     NET.TIME(TRAVEL.REQUEST),
     DAY,
     NLT.TIME(TRAVEL.REQUEST),
     PAX.LOAD(TRAVEL.REQUEST),
     PAX.PRIORITY(TRAVEL.REQUEST),
     DV.CODE(TRAVEL.REQUEST),
     PAX.1NAME(TRAVEL.REQUEST),
     PAX.2NAME(TRAVEL.REQUEST),
     PAX.RANK(TRAVEL.REQUEST)
     AS S 1,A 8,A 7,I 3,D(6,2),I 4,D(6,2),I 3,I 4,I 7,S 3,A 10,A 7,A 4,/
RETURN
END ''OF READ.DATA
```

```
ROUTINE T39.SCHEDULER

DEFINE
     ICID,
     SELECTED,
     NAME,
     SBLOC,
     SBDEST,
     LPTOON,
     CK.BASE,
     CK.HOME
     AS ALPHA VARIABLES
DEFINE
     REQ1,
     BASE1,
     HOP1,HOP2,HOP3,
     REQ2,
     CK.MAX,
     THRUSEATS,
     INSEATS,
     OUTSEATS,
     PAXTHRU,
     PAXIN,
     PAXOUT,
     RF1.BASE,RF2.BASE,
     REQ3,
     BSTR3,
     LEG1,LEG2,LEG3,LEG4,LEG5,LEG7,LEG8,
     LEG35,LEG36, LEG37,LEG38,
     SIEVE,
     SEPAX,
     NUMPAX
     AS INTEGER VARIABLES
SKIP 2 LINES
PRINT 1 LINE THUS
```

```
REQUESTS WERE SCHEDULED IN THIS ORDER AND LOADED ON INDICATED SABRELINERS
SKIP 1 LINE
PRINT 1 DOUBLE LINE THUS
#T39 LOCATION  TERM PT       NAME                    ORIGIN DEST PRI DVC RANK
SELECTED
SKIP 1 LINE
'SKED'
''CHECK TO SEE IF ALL REQUESTS HAVE BEEN SATISFIED OR
''IF ALL AIRPLANES HAVE BEEN SCHEDULED
IF N.UNSATISFIED.REQUESTS EQ 0 OR T39S.AVAILABLE EQ 0
     GO TO INFEASIBLE
ELSE
'' ASSIGN 0-10S TO CLOSEST AVAILABLE SABRELINER AND LIST REQUESTS FOR FOR
''MANUAL SCHEDULING
FOR EACH REQ1 IN UNSATISFIED.REQUESTS WITH PAX.RANK(REQ1) EQ "00/A"
     FIND THE FIRST CASE
IF FOUND
     REMOVE REQ1 FROM UNSATISFIED.REQUESTS
     LET SELECTED="SCHED MAN"
     GO TO S3
ELSE
''SPECFIC REQUESTS ARE ALIGNED TO SPECFIC SABRELINERS BY ADDING ONE DATA
''CARD WITH THE FOLLOWING INFORMATION: FIRST 10 LETTERS OF PRIORITY
''THREE'S PAX.1NAME AS IT APPEARS ON THE REQUEST IN COLUMNS 1-10,
''ICAO IDENTIFIER OF SABRELINER LOCATION IN COLUMNS 11-14, AND ICAO
''IDENTIFIER OF SABRELINER DESTINATION IN COLUMNS 15-18  EXAMPLE:
''        JONES, HOWKADWKMCC
''WITH SABRELINER LOCATION AND DESTINATION LEFT BLANK, REQUEST WILL
''BE SCHEDULED NORMALLY IN THE SEQUENCE ENTERED.
'S5'
IF DATA IS ENDED
     LET SELECTED="ALGORITHM"
     GO TO S1
ELSE
     LET SELECTED="SCHEDULER"
START NEW CARD
```

```
READ NAME,SBLOC,SBDEST AS A 10,A 4,A 5
     FOR EACH REQ1 IN UNSATISFIED.REQUESTS WITH PAX.1NAME(REQ1) EQ
          NAME AND
          PAX.PRIORITY(REQ1) LE 3
     FIND THE FIRST CASE
          IF NONE
               GO TO S5
          ELSE
          IF SBLOC EQ "     "
               REMOVE REQ1 FROM UNSATISFIED.REQUESTS
               GO TO S3
          ELSE
     FOR EACH BASE IN BASE.FILE WITH ICAO(BASE) EQ SBLOC
          AND N.DET(BASE) GT 0
     DO
          FOR EACH SABRELINER IN DET(BASE) WITH DUTY.DAY(SABRELINER) EQ
          0 AND HOME.STATION(SABRELINER) EQ SBDEST
               FIND THE FIRST CASE
               IF NONE
                    GO TO S4
               ELSE
          REMOVE REQ1 FROM UNSATISFIED.REQUESTS
               GO TO S2
'S4'
     LOOP
     GO TO S1
'S2'
     IF NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE)-2 GE 6 AND
          NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE)-2 LE 10
          LET MAX=14
          LET CK.MAX=14
     ELSE
          LET MAX=12
          LET CK.MAX=12
     ALWAYS
CALL LEG.DATA GIVEN R.ORIGIN(REQ1),R.DESTINATION(REQ1)
```

```
    YIELDING DIST1,HOP1,TIME1,FUEL1,GS1,TC1,LAT1,LONG1
IF R.ORIGIN(REQ1) NE ICAO(BASE)
    CALL LEG.DATA GIVEN ICAO(BASE),R.ORIGIN(REQ1)
        YIELDING DIST2,HOP2,TIME2,FUEL2,GS2,TC2,LAT2,LONG2
    CALL LEG.DATA GIVEN R.DESTINATION(REQ1),HOME.STATION(SABRELINER)
        YIELDING DIST3,HOP3,TIME3,FUEL3,GS3,TC3,LAT3,LONG3
    IF NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE)-TIME2-3 GE 6 AND
        NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE)-3-TIME2 LE 10
        LET MAX=14
        LET CK.MAX=14
    ELSE
        LET MAX=12
        LET CK.MAX=12
    ALWAYS
    GO TO OPT3
ELSE
    IF HOP1 EQ 0
        GO TO OPT1
    ELSE
        GO TO OPT2
'S1'
''ASSIGN AND SCHEDULE REQUESTS WHICH MATCH ONE FOR ONE WITH SABRELINER'S
''DEPARTURE AND DESTINATION(RON POINT)
FOR EACH BASE IN BASE.FILE WITH N.DET(BASE) GT 0
DO
    FOR EACH SABRELINER IN DET(BASE) WITH DUTY.DAY(SABRELINER) EQ 0
        AND ICAO(BASE) NE HOME.STATION(SABRELINER)
        FIND THE FIRST CASE
            IF NONE
                GO TO L1
            ELSE
                FOR EACH REQ1 IN UNSATISFIED.REQUESTS WITH
                PAX.PRIORITY(REQ1) LE 3 AND R.ORIGIN(REQ1) EQ
                    ICAO(BASE)
                AND  R.DESTINATION(REQ1) EQ HOME.STATION(SABRELINER)
                FIND THE FIRST CASE
```

```
                    IF FOUND
                    REMOVE REQ1 FROM UNSATISFIED.REQUESTS
                    GO TO S3
                    ELSE
'L1'
LOOP
''CHECK FOR AN INTERPLANE MISSION
''IF NONE, ATTEMPT TO SCHEDULE THE PRIORITY 1, 2, OR 3 REQUEST WITH THE
''EARLIEST NET TIME, THEN THE HIGHEST PRIORITY MISSION WITH FIVE
''PASSENGERS, THEN FOUR, ETC. THAT DOES NOT REQUIRE AN INTERPLANE.
FOR EACH REQ1 IN UNSATISFIED.REQUESTS,WITH PAK.2NAME(REQ1) EQ "XXXXX"
     AND PAX.PRIORITY(REQ1) LT 20,
     FIND THE FIRST CASE
     IF FOUND
          REMOVE REQ1 FROM UNSATISFIED.REQUESTS
     ELSE
          REMOVE FIRST REQ1 FROM UNSATISFIED.REQUESTS
          IF PAX.PRIORITY(REQ1) GT 3
               FILE REQ1 IN UNSATISFIED.REQUESTS
FOR EACH BASE IN BASE.FILE WITH N.DET(BASE) GT 0
DO
     FOR EACH SABRELINER IN DET(BASE) WITH DUTY.DAY(SABRELINER) EQ 0
          FIND THE FIRST CASE
          IF NONE
               GO TO S7
          ELSE
               LET NUMPAX=5
               HERE
               FOR EACH REQ1 IN UNSATISFIED.REQUESTS
                    WITH PAX.LOAD(REQ1) EQ NUMPAX
                    AND PAX.PRIORITY(REQ1) LE 20
                    AND R.ORIGIN(REQ1) EQ ICAJ(BASE)
                    FIND THE FIRST CASE
                    IF NONE
                         LET NUMPAX=NUMPAX-1
                         IF NUMPAX GT 2
```

```
                                    JUMP BACK
                               ELSE
                                    LET NUMPAX=5
                                    HERE
                                    FOR EACH REQ1 IN UNSATISFIED.REQUESTS
                                         WITH PAX.LOAD(REQ1) EQ NUMPAX
                                         AND PAX.PRIORITY(REQ1) LE 20
                                         FIND THE FIRST CASE
                                         IF NONE
                                              LET NJMPAX=NUMPAX-1
                                              IF NUMPAX GT 0
                                                   JUMP BACK
                                              ELSE
                                                   GO TO INFEASIBLE
                                         ELSE
                                         REMOVE REQ1 FROM UNSATISFIED.REQUESTS
                                         GO TO S3
                          ELSE
                          REMOVE REQ1 FROM UNSATISFIED.REQUESTS
                          GO TO S3
'S7'
LOOP
          ALWAYS
     ALWAYS
'S3'
''CHECK TO SEE IF ALL REMAINING REQUESTS ARE INFEASIBLE
IF PAX.PRIORITY(REQ1) GT 20
     FILE REQ1 IN UNSATISFIED.REQUESTS
     GO TO INFEASIBLE
ELSE
CALL LEG.DATA GIVEN R.ORIGIN(REQ1),R.DESTINATION(REQ1)
     YIELDING DIST1,HOP1,TIME1,FUEL1,GS1,TC1,LAT1,LONG1
''CHECK FOR AN AIRPLANE AT THE ORIGIN THAT CAN SUPPORT THE REQUEST
''WITHIN CREW DUTY DAY LIMITATIONS
FOR EACH BASE IN BASE.FILE,WITH ICAO(BASE) EQ R.ORIGIN(REQ1)
DO
```

```
    IF N.DET(BASE) EQ 0
        JUMP AHEAD
    ELSE
    IF NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE)-2 GE 6 AND
        NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE)-2 LE 10
        LET MAX=14
    ELSE
        LET MAX=12
    ALWAYS
    FOR EACH SABRELINER IN DET(BASE),WITH DUTY.DAY(SABRELINER) EQ 0,
        FIND THE FIRST CASE
        IF NONE
            JUMP AHEAD
        ELSE
        IF R.DESTINATION(REQ1) EQ HOME.STATION(SABRELINER)
            IF TIME1 GT MAX-2
                JUMP AHEAD
            ELSE
            IF HOP1 EQ 0
                GO TO OPT1
            ELSE
                GO TO OPT2
        ELSE
            CALL LEG.DATA GIVEN R.DESTINATION(REQ1),
                HOME.STATION(SABRELINER)
                YIELDING DIST2,HOP2,TIME2,FUEL2,GS2,TC2,LAT2,LONG2
            IF TIME1+TIME2 GT MAX-3
                JUMP AHEAD
            ELSE
            IF HOP1 EQ 0
                GO TO OPT1
            ELSE
                GO TO OPT2
    HERE
LOOP
''IF THERE ARE NO AIRPLANES AT THE ORIGIN THAT CAN SUPPORT THE REQUEST
```

```
''WITHIN CREW DUTY DAY LIMITATIONS, CHECK FOR AIRPLANES AT OTHER
''LOCATIONS THAT CAN SUPPORT THE REQUEST WITHIN CREW DUTY DAY
''LIMITATIONS
LET CK.PAD=0
FOR EACH BASE IN BASE.FILE,WITH
     N.DET(BASE) GT 0 AND
     ICAO(BASE) NE R.ORIGIN(REQ1)
DO
     FOR EACH SABRELINER IN DET(BASE),WITH DUTY.DAY(SABRELINER) EQ 0,
          FIND THE FIRST CASE
          IF NONE
               JUMP AHEAD
          ELSE
          CALL LEG.DATA GIVEN ICAO(BASE),R.ORIGIN(REQ1)
               YIELDING DIST2,HOP2,TIME2,FUEL2,GS2,TC2,LAT2,LONG2
               IF NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE)-TIME2-3 GE 6 AND
                    NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE)-TIME2-3 LE 10
                    LET MAX=14
               ELSE
                    LET MAX=12
               ALWAYS
               IF R.DESTINATION(REQ1) EQ HOME.STATION(SABRELINER)
                    LET PAD=MAX-TIME1-TIME2-3
                    IF PAD GT CK.PAD
                         LET CK.PAD=PAD
                         LET CK.BASE=ICAO(BASE)
                         LET CK.HOME=HOME.STATION(SABRELINER)
                         LET CK.MAX=MAX
                    ALWAYS
               ELSE
                    CALL LEG.DATA GIVEN R.DESTINATION(REQ1),
                         HOME.STATION(SABRELINER)
                         YIELDING X,X,TIME3
                         LET PAD=MAX-TIME1-TIME2-TIME3-4
                         IF PAD GT CK.PAD
                              LET CK.PAD=PAD
```

```
                                LET CK.BASE=ICAO(BASE)
                                LET CK.HOME=HOME.STATION(SABRELINER)
                                LET CK.MAX=MAX
                            ALWAYS
                    ALWAYS
                    HERE
LOOP
IF CK.PAD EQ 0
''IF NO SINGLE AIRPLANE CAN SUPPORT THE REQUEST WITHIN CREW DUTY DAY
''LIMITATIONS, CHECK TO SEE IF TWO AIRPLANES CAN SUPPORT THE REQUEST
''WITHIN THEIR CREW DUTY DAY LIMITATIONS
    LET IPRQPT=REQ1
    FILE REQ1 IN UNSATISFIED.REQUESTS
    CALL INTERPLANE
    GO TO SKED
ELSE
    FOR EACH BASE IN BASE.FILE,WITH ICAO(BASE) EQ CK.BASE
    DO
        FOR EACH SABRELINER IN DET(BASE),WITH
            DUTY.DAY(SABRELINER) EQ 0 AND
            HOME.STATION(SABRELINER) EQ CK.HOME,
            FIND THE FIRST CASE
            IF FOUND
                JUMP AHEAD
            ELSE
    LOOP
    HERE
    CALL LEG.DATA GIVEN ICAO(BASE),R.ORIGIN(REQ1)
        YIELDING DIST2,HOP2,TIME2,FUEL2,GS2,TC2,LAT2,LONG2
    CALL LEG.DATA GIVEN R.DESTINATION(REQ1),HOME.STATION(SABRELINER)
        YIELDING DIST3,HOP3,TIME3,FUEL3,GS3,TC3,LAT3,LONG3
    GO TO OPT3
'OPT1'
''AIRCRAFT AT ORIGIN, REFUELING NOT REQUIRED
''BETWEEN ORIGIN AND DESTINATION
IF MAX.DUTY.DAY(SABRELINER) LT  10
```

```
        LET CREW.DUTY.START(SABRELINER)=NET.TIME(REQ1)-2
        LET MAX.DUTY.DAY(SABRELINER)=MAX
ALWAYS
LET SEATS.AVAILABLE(SABRELINER)=5-PAX.LOAD(REQ1)
'OPT2A'
''ENTRY POINT FROM OPT2
''AIRCRAFT AT ORIGIN, REFUELING REQUIRED BETWEEN ORIGIN AND DESTINATION
''HOWEVER, REFUELING BASE NOT IDENTIFIED BY PROGRAM
CREATE A LEG CALLED LEG4
LET L.ORIGIN(LEG4)=R.ORIGIN(REQ1)
LET L.DESTINATION(LEG4)=R.DESTINATION(REQ1)
LET DEPARTURE.TIME(LEG4)=NET.TIME(REQ1)
LET ENROUTE.TIME(LEG4)=TIME1
LET ARRIVAL.TIME(LEG4)=DEPARTURE.TIME(LEG4)+ENROUTE.TIME(LEG4)
LET FUEL.CONSUMED(LEG4)=FUEL1
FILE REQ1 IN SATISFIED.REQUESTS(LEG4)
FILE LEG4 IN ITINERARY(SABRELINER)
''SEARCH FOR ADDITIONAL PASSENGERS
'1ASEARCH'
IF SEATS.AVAILABLE(SABRELINER) GT 0
        FOR EACH REQ IN UNSATISFIED.REQUESTS, WITH
                R.ORIGIN(REQ) EQ L.ORIGIN(LEG4) AND
                R.DESTINATION(REQ) EQ L.DESTINATION(LEG4) AND
                NET.TIME(REQ) LE DEPARTURE.TIME(LEG4) AND
                NLT.TIME(REQ) GE ARRIVAL.TIME(LEG4) AND
                PAX.LOAD(REQ) LE SEATS.AVAILABLE(SABRELINER),
                FIND THE FIRST CASE
                IF FOUND
                        REMOVE REQ FROM UNSATISFIED.REQUESTS
                        FILE REQ IN SATISFIED.REQUESTS(LEG4)
                        LET SEATS.AVAILABLE(SABRELINER)=
                                SEATS.AVAILABLE(SABRELINER)-PAX.LOAD(REQ)
                        GO TO 1ASEARCH
                ELSE
ALWAYS
GO TO CONTINUE
```

```
'OPT2'
''AIRCRAFT AVAILABLE AT ORIGIN, REFUELING REQUIRED
''BETWEEN ORIGIN AND DESTINATION
IF MAX.DUTY.DAY(SABRELINER) LT  10
     LET CREW.DUTY.START(SABRELINER)=NET.TIME(REQ1)-2
     LET MAX.DUTY.DAY(SABRELINER)=MAX
ALWAYS
LET SEATS.AVAILABLE(SABRELINER)=5-PAX.LOAD(REQ1)
CALL REFUEL GIVEN R.ORIGIN(REQ1),R.DESTINATION(REQ1),DIST1,GS1,TC1,LAT1,
     LONG1,0,0
''FEASIBILITY CHECK
IF N.REF.BASE.FILE EQ 0
     GO TO OPT2A
ELSE
''FEASIBILITY CHECK
FOR EACH RF.BASE IN REF.BASE.FILE, WITH TOT.TIME(RF.BASE) LE TIME1+.5,
     FIND THE FIRST CASE
     IF NONE
          GO TO OPT2A
     ELSE
CREATE A LEG CALLED LEG3
LET L.ORIGIN(LEG3)=R.ORIGIN(REQ1)
LET L.DESTINATION(LEG3)=RF.ICAO(RF.BASE)
LET DEPARTURE.TIME(LEG3)=NET.TIME(REQ1)
LET ENROUTE.TIME(LEG3)=IN.TIME(RF.BASE)
LET ARRIVAL.TIME(LEG3)=DEPARTURE.TIME(LEG3)+ENROUTE.TIME(LEG3)
CREATE A LEG CALLED LEG4
LET FUEL.CONSUMED(LEG3)=IN.FUEL(RF.BASE)
FILE REQ1 IN SATISFIED.REQUESTS(LEG3)
FILE LEG3 IN ITINERARY(SABRELINER)
LET L.ORIGIN(LEG4)=RF.ICAO(RF.BASE)
LET L.DESTINATION(LEG4)=R.DESTINATION(REQ1)
LET DEPARTURE.TIME(LEG4)=ARRIVAL.TIME(LEG3)+1
LET ENROUTE.TIME(LEG4)=OUT.TIME(RF.BASE)
LET ARRIVAL.TIME(LEG4)=DEPARTURE.TIME(LEG4)+ENROUTE.TIME(LEG4)
LET FUEL.CONSUMED(LEG4)=OUT.FUEL(RF.BASE)
```

```
FILE LEG4 IN ITINERARY(SABRELINER)
IF SEATS.AVAILABLE(SABRELINER) GT 0
     LET PAXTHRU=0
     LET PAXIN=0
     LET PAXOUT=0
     LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)
     LET INSEATS=SEATS.AVAILABLE(SABRELINER)
     LET OUTSEATS=SEATS.AVAILABLE(SABRELINER)
''SEARCH FOR ADDITIONAL PASSENGERS
'1BSEARCH'
IF THRUSEATS GT 0 OR OUTSEATS GT 0 OR INSEATS GT 0
     FOR EACH REQ IN UNSATISFIED.REQUESTS,WITH
          NET.TIME(REQ) LE DEPARTURE.TIME(LEG3) AND
          (R.ORIGIN(REQ) EQ L.ORIGIN(LEG3) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG4) AND
          NLT.TIME(REQ) GE ARRIVAL.TIME(LEG4) AND
          PAX.LOAD(REQ) LE THRUSEATS) OR
          (R.ORIGIN(REQ) EQ L.ORIGIN(LEG3) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG3) AND
          NET.TIME(REQ) LE DEPARTURE.TIME(LEG3) AND
          NLT.TIME(REQ) GE ARRIVAL.TIME(LEG3) AND
          PAX.LOAD(REQ) LE INSEATS) OR
          (R.ORIGIN(REQ) EQ L.ORIGIN(LEG4) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG4) AND
          NET.TIME(REQ) LE DEPARTURE.TIME(LEG4) AND
          NLT.TIME(REQ) GE ARRIVAL.TIME(LEG4) AND
          PAX.LOAD(REQ) LE OUTSEATS)
          FIND THE FIRST CASE
          IF FOUND
               REMOVE REQ FROM UNSATISFIED.REQUESTS
               IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG3) AND
                    R.DESTINATION(REQ) EQ L.DESTINATION(LEG4)
                    LET PAXTHRU=PAXTHRU+PAX.LOAD(REQ)
                    LET THRUSEATS=THRUSEATS-PAX.LOAD(REQ)
                    LET INSEATS=INSEATS-PAX.LOAD(REQ)
                    LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ)
```

```
            FILE REQ IN SATISFIED.REQUESTS(LEG3)
        ALWAYS
        IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG3) AND
            R.DESTINATION(REQ) EQ L.DESTINATION(LEG3)
            LET PAXIN=PAXIN+PAX.LOAD(REQ)
            LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
                MAX.F(PAXIN,PAXOUT)
            LET INSEATS=INSEATS-PAX.LOAD(REQ)
            FILE REQ IN SATISFIED.REQUESTS(LEG3)
        ALWAYS
        IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG4) AND
            R.DESTINATION(REQ) EQ L.DESTINATION(LEG4)
            LET PAXOUT=PAXOUT+PAX.LOAD(REQ)
            LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
                MAX.F(PAXIN,PAXOUT)
            LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ)
            FILE REQ IN SATISFIED.REQUESTS(LEG4)
        ALWAYS
        GO TO 1BSEARCH
    ELSE
ALWAYS
ALWAYS
GO TO CONTINUE
'OPT3'
''AIRCRAFT NOT AVAILABLE AT ORIGIN
LET CREW.DUTY.START(SABRELINER)=NET.TIME(REQ1)-(3+TIME2)
''AS MUCH AS 30 MINUTES IN ERROR IF REFUELING IS REQUIRED TO REACH
''THE ORIGIN
LET MAX.DUTY.DAY(SABRELINER)=CK.MAX
IF HOP2 GT 0 ''REFUELING REQUIRED TO REACH ORIGIN
CALL REFUEL GIVEN ICAO(BASE),R.ORIGIN(REQ1),DIST3,GS3,TC3,LAT3,LONG3,0,0
''FEASIBILITY CHECK
IF N.REF.BASE.FILE EQ 0
    GO TO OPT3A
ELSE
''FEASIBILITY CHECK
```

```
            FILE REQ IN SATISFIED.REQUESTS(LEG3)
        ALWAYS
        IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG3) AND
            R.DESTINATION(REQ) EQ L.DESTINATION(LEG3)
            LET PAXIN=PAXIN+PAX.LOAD(REQ)
            LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
                MAX.F(PAXIN,PAXOUT)
            LET INSEATS=INSEATS-PAX.LOAD(REQ)
            FILE REQ IN SATISFIED.REQUESTS(LEG3)
        ALWAYS
        IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG4) AND
            R.DESTINATION(REQ) EQ L.DESTINATION(LEG4)
            LET PAXOUT=PAXOUT+PAX.LOAD(REQ)
            LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
                MAX.F(PAXIN,PAXOUT)
            LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ)
            FILE REQ IN SATISFIED.REQUESTS(LEG4)
        ALWAYS
        GO TO 1BSEARCH
    ELSE
ALWAYS
ALWAYS
GO TO CONTINUE
'OPT3'
''AIRCRAFT NOT AVAILABLE AT ORIGIN
LET CREW.DUTY.START(SABRELINER)=NET.TIME(REQ1)-(3+TIME2)
''AS MUCH AS 30 MINUTES IN ERROR IF REFUELING IS REQUIRED TO REACH
''THE ORIGIN
LET MAX.DUTY.DAY(SABRELINER)=CK.MAX
IF HOP2 GT 0 ''REFUELING REQUIRED TO REACH ORIGIN
CALL REFUEL GIVEN ICAO(BASE),R.ORIGIN(REQ1),DIST2,GS2,TC2,LAT2,LONG2
''FEASIBILITY CHECK
IF N.REF.BASE.FILE EQ 0
    GO TO OPT3A
ELSE
''FEASIBILITY CHECK
```

```
FOR EACH RF.BASE IN REF.BASE.FILE, WITH TOT.TIME(RF.BASE) LE TIME2+.5
     FIND THE FIRST CASE
     IF NONE
          GO TO OPT3A
     ELSE
CREATE A LEG CALLED LEG1
LET L.ORIGIN(LEG1)=ICAO(BASE)
LET L.DESTINATION(LEG1)=RF.ICAO(RF.BASE)
LET DEPARTURE.TIME(LEG1)=NET.TIME(REQ1)-(1+TOT.TIME(RF.BASE))
LET ENROUTE.TIME(LEG1)=IN.TIME(RF.BASE)
LET ARRIVAL.TIME(LEG1)=DEPARTURE.TIME(LEG1)+ENROUTE.TIME(LEG1)
LET FUEL.CONSUMED(LEG1)=IN.FUEL(RF.BASE)
FILE LEG1 IN ITINERARY(SABRELINER)
CREATE A LEG CALLED LEG2
LET L.ORIGIN(LEG2)=RF.ICAO(RF.BASE)
LET L.DESTINATION(LEG2)=R.ORIGIN(REQ1)
LET DEPARTURE.TIME(LEG2)=ARRIVAL.TIME(LEG1)+1
LET ENROUTE.TIME(LEG2)=OUT.TIME(RF.BASE)
LET ARRIVAL.TIME(LEG2)=DEPARTURE.TIME(LEG2)+ENROUTE.TIME(LEG2)
LET FUEL.CONSUMED(LEG2)=OUT.FUEL(RF.BASE)
FILE LEG2 IN ITINERARY(SABRELINER)
IF SEATS.AVAILABLE(SABRELINER) GT 0
     LET PAXTHRU=0
     LET PAXIN=0
     LET PAXOUT=0
     LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)
     LET INSEATS=SEATS.AVAILABLE(SABRELINER)
     LET OUTSEATS=SEATS.AVAILABLE(SABRELINER)
''SEARCH FOR PASSENGERS
'1DSEARCH'
IF THRUSEATS GT 0 OR OUTSEATS GT 0 OR INSEATS GT 0
     FOR EACH REQ IN UNSATISFIED.REQUESTS,WITH
          (R.ORIGIN(REQ) EQ L.ORIGIN(LEG1) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG2) AND
          NET.TIME(REQ) LE DEPARTURE.TIME(LEG1) AND
          NLT.TIME(REQ) GE ARRIVAL.TIME(LEG2) AND
```

```
PAX.LOAD(REQ) LE THRUSEATS) OR
(R.ORIGIN(REQ) EQ L.ORIGIN(LEG1) AND
R.DESTINATION(REQ) EQ L.DESTINATION(LEG1) AND
NET.TIME(REQ) LE DEPARTURE.TIME(LEG1) AND
NLT.TIME(REQ) GE ARRIVAL.TIME(LEG1) AND
PAX.LOAD(REQ) LE INSEATS) OR
(R.ORIGIN(REQ) EQ L.ORIGIN(LEG2) AND
R.DESTINATION(REQ) EQ L.DESTINATION(LEG2) AND
NET.TIME(REQ) LE DEPARTURE.TIME(LEG2) AND
NLT.TIME(REQ) GE ARRIVAL.TIME(LEG2) AND
PAX.LOAD(REQ) LE OUTSEATS)
FIND THE FIRST CASE
IF FOUND
     REMOVE REQ FROM UNSATISFIED.REQUESTS
     IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG1) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG2)
          LET PAXTHRU=PAXTHRU+PAX.LOAD(REQ)
          LET THRUSEATS=THRUSEATS-PAX.LOAD(REQ)
          LET INSEATS=INSEATS-PAX.LOAD(REQ)
          LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ)
          FILE REQ IN SATISFIED.REQUESTS(LEG1)
     ALWAYS
     IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG1) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG1)
          LET PAXIN=PAXIN+PAX.LOAD(REQ)
          LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
               MAX.F(PAXIN,PAXOUT)
          LET INSEATS=INSEATS-PAX.LOAD(REQ)
          FILE REQ IN SATISFIED.REQUESTS(LEG1)
     ALWAYS
     IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG2) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG2)
          LET PAXOUT=PAXOUT+PAX.LOAD(REQ)
          LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
               MAX.F(PAXIN,PAXOUT)
          LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ)
```

```
                FILE REQ IN SATISFIED.REQUESTS(LEG2)
            ALWAYS
            GO TO 1DSEARCH
        ELSE
ALWAYS
ALWAYS
ELSE ''REFUELING NOT REQUIRED TO REACH ORIGIN
'OPT3A'
''ENTRY POINT FROM OPT3 ABOVE
''AIRCRAFT NOT AVAILABLE AT ORIGIN, REFUELING REQUIRED BETWEEN
''ORIGIN AND DESTINATION
''HOWEVER,REFUELING BASE NOT IDENTIFIED BY PROGRAM
CREATE A LEG CALLED LEG2
LET L.ORIGIN(LEG2)=ICAO(BASE)
LET L.DESTINATION(LEG2)=R.ORIGIN(REQ1)
LET DEPARTURE.TIME(LEG2)=NET.TIME(REQ1)-(1+TIME2)
LET ENROUTE.TIME(LEG2)=TIME2
LET ARRIVAL.TIME(LEG2)=DEPARTURE.TIME(LEG2)+ENROUTE.TIME(LEG2)
LET FUEL.CONSUMED(LEG2)=FUEL2
FILE LEG2 IN ITINERARY(SABRELINER)
'4DSEARCH'
IF SEATS.AVAILABLE(SABRELINER) GT 0
    FOR EACH REQ IN UNSATISFIED.REQUESTS,WITH
        R.ORIGIN(REQ) EQ L.ORIGIN(LEG2) AND
        R.DESTINATION(REQ) EQ L.DESTINATION(LEG2) AND
        NET.TIME(REQ) LE DEPARTURE.TIME(LEG2) AND
        NLT.TIME(REQ) GE ARRIVAL.TIME(LEG2) AND
        PAX.LOAD(REQ) LE SEATS.AVAILABLE(SABRELINER),
        FIND THE FIRST CASE
        IF FOUND
            REMOVE REQ FROM UNSATISFIED.REQUESTS
            FILE REQ IN SATISFIED.REQUESTS(LEG2)
            LET SEATS.AVAILABLE(SABRELINER)=
                SEATS.AVAILABLE(SABRELINER)-PAX.LOAD(REQ)
            GO TO 4DSEARCH
        ELSE
```

```
ALWAYS
ALWAYS ''SCHEDULE MISSION FROM ORIGIN TO DESTINATION
IF HOP1 EQ 0
     GO TO OPT1
ELSE
     GO TO OPT2
'CONTINUE'
''SABRELINER IS NOW AT DESTINATION WITH ENOUGH CREW DUTY TIME REMAINING
''TO RETURN TO HOME STATION
IF CT39 NE T39S.AVAILABLE
''WRITE AIRCRAFT'S DEPARTURE POINT AND DESTINATION AND REQUESTER'S NAME.
WRITE T39S.AVAILABLE,
     ICAO(BASE),
     HOME.STATION(SABRELINER),
     PAX.1NAME(REQ1),
     PAX.2NAME(REQ1),
     R.ORIGIN(REQ1),
     R.DESTINATION(REQ1),
     PAX.PRIORITY(REQ1),
     DV.CODE(REQ1),
     PAX.RANK(REQ1),
     SELECTED
     AS S 1,I 2,S 4,A 4, S 5,A 4,S 9,A 10,A 7,S 4,A 4,S 2,A 4,S 1,I 2,
          S 3,I 1,S 2,A 4,S 6,A 9,/
LET CT39=T39S.AVAILABLE
ALWAYS
LET DUTY.DAY(SABRELINER)= ARRIVAL.TIME(L.ITINERARY(SABRELINER))+1-
     CREW.DUTY.START(SABRELINER)
IF PAX.RANK(REQ1) EQ "00/A"
     LET ICID=R.DESTINATION(REQ1)
     HERE
     FOR EACH REQ2 IN UNSATISFIED.REQUESTS WITH
          PAX.1NAME(REQ2) EQ PAX.1NAME(REQ1) AND
          PAX.2NAME(REQ2) EQ PAX.2NAME(REQ1) AND
          PAX.RANK(REQ2) EQ PAX.RANK(REQ1) AND
          R.ORIGIN(REQ2) EQ ICID
```

```
    FIND THE FIRST CASE
    IF FOUND
        CREATE A LEG
        LET L.ORIGIN(LEG)=R.ORIGIN(REQ2)
        LET DEPARTURE.TIME(LEG)=NET.TIME(REQ2)
        LET L.DESTINATION(LEG)=R.DESTINATION(REQ2)
        LET ARRIVAL.TIME(LEG)=NLT.TIME(REQ2)
        REMOVE REQ2 FROM UNSATISFIED.REQUESTS
        FILE REQ2 IN SATISFIED.REQUESTS(LEG)
        FILE LEG IN ITINERARY(SABRELINER)
        LET ICID=R.DESTINATION(REQ2)
        JUMP BACK
    ELSE
        GO TO TERM
ELSE
''MATCH ANY MANUALLY SPLIT REQUESTS.
FOR EACH BSTR3 IN UNSATISFIED.REQUESTS WITH PAX.2NAME(BSTR3) EQ "QQQQQ"
    AND R.ORIGIN(BSTR3) EQ L.DESTINATION(L.ITINERARY(SABRELINER))
    AND NET.TIME(BSTR3) GE ARRIVAL.TIME(L.ITINERARY(SABRELINER))
    FIND THE FIRST CASE
    IF FOUND
        REMOVE BSTR3 FROM UNSATISFIED.REQUESTS
        CALL LEG.DATA GIVEN R.ORIGIN(BSTR3),R.DESTINATION(BSTR3)
            YIELDING DIST1,HOP1,TIME1,FUEL1,GS1,TC1,LAT1,LONG1
        CALL LEG.DATA GIVEN L.DESTINATION(L.ITINERARY(SABRELINER)),
            R.ORIGIN(BSTR3) YIELDING DIST2,HOP2,TIME2,FUEL2,GS2,TC2,
                LAT2,LONG2
        CALL LEG.DATA GIVEN R.DESTINATION(BSTR3),
            HOME.STATION(SABRELINER) YIELDING DIST3,HOP3,TIME3,
                FUEL3,GS3,TC3,LAT3,LONG3
        GO TO C1
    ELSE
''SEARCH FOR PRIORITY 3 REQUEST THAT CAN BE SATISFIED WITHIN CREW DAY.
''IF NONE, SEARCH FOR HIGHEST PRIORITY REQUEST WITH # OF PAX=5,
''     THEN 4, 3, ETC.
LET EPTO= ARRIVAL.TIME(L.ITINERARY(SABRELINER))+1
```

```
LET LPTERM=MAX.DUTY.DAY(SABRELINER)+CREW.DUTY.START(SABRELINER)
LET NUMPAX=0
LET SEPAX=1
LET SIEVE=3
'SR5'
FOR EACH BSTR3 IN UNSATISFIED.REQUESTS WITH
      PAX.PRIORITY(BSTR3) LE SIEVE AND
      PAX.LOAD(BSTR3) GE SEPAX AND
      NET.TIME(BSTR3) LT LPTERM-.5 AND
      NLT.TIME(BSTR3) GT EPTO+.5
      DO
            CALL LEG.DATA GIVEN R.ORIGIN(BSTR3),R.DESTINATION(BSTR3)
                  YIELDING DIST1,HOP1,TIME1,FUEL1,GS1,TC1,LAT1,LONG1
            CALL LEG.DATA GIVEN L.DESTINATION(L.ITINERARY(SABRELINER)),
                  R.ORIGIN(BSTR3) YIELDING DIST2,HOP2,TIME2,FUEL2,GS2,TC2,
                        LAT2,LONG2
            CALL LEG.DATA GIVEN R.DESTINATION(BSTR3),
                  HOME.STATION(SABRELINER) YIELDING DIST3,HOP3,TIME3,
                        FUEL3,GS3,TC3,LAT3,LONG3
            LET EPTOA=EPTO+TIME2+TIME2/(TIME2+.000001)
            LET EPLND=EPTOA+TIME1
            LET LPLND=LPTERM-TIME3-TIME3/(TIME3+.000001)
            LET LPTO=LPLND-TIME1
            IF NET.TIME(BSTR3) LE LPTO AND
                  NLT.TIME(BSTR3) GE EPLND AND
                  TIME1 LT LPLND-MAX.F(NET.TIME(BSTR3),EPTOA)
                  GO TO ROUT2
            ELSE
                  GO TO ROUT1
'ROUT1'
      LOOP
            LET SIEVE=12
            LET NUMPAX=NUMPAX+1
            IF NUMPAX EQ 6
                  GO TO RCONTINUE
            ELSE
```

```
                LET SEPAX=6-NUMPAX
                GO TO SR5
'ROUT2'
     REMOVE BSTR3 FROM UNSATISFIED.REQUESTS
'C1'
LET SEATS.AVAILABLE(SABRELINER)=5
IF TIME2 GT 0
     IF HOP2 EQ 0
          CREATE A LEG CALLED LEG36
          LET L.ORIGIN(LEG36)=L.DESTINATION(L.ITINERARY(SABRELINER))
          LET L.DESTINATION(LEG36)=R.ORIGIN(BSTR3)
          LET DEPARTURE.TIME(LEG36)=ARRIVAL.TIME(L.ITINERARY(SABRELINER))+1
          LET ENROUTE.TIME(LEG36)=TIME2
          LET ARRIVAL.TIME(LEG36)=DEPARTURE.TIME(LEG36)+ENROUTE.TIME(LEG36)
          LET FUEL.CONSUMED(LEG36)=FUEL2
          FILE LEG36 IN ITINERARY(SABRELINER)
''SEARCH FOR ADDITIONAL PASSENGERS
HERE
IF SEATS.AVAILABLE(SABRELINER) GT 0
     FOR EACH REQ IN UNSATISFIED.REQUESTS, WITH
          R.ORIGIN(REQ) EQ L.ORIGIN(LEG36) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG36) AND
          NET.TIME(REQ) LE DEPARTURE.TIME(LEG36) AND
          NLT.TIME(REQ) GE ARRIVAL.TIME(LEG36) AND
          PAX.LOAD(REQ) LE SEATS.AVAILABLE(SABRELINER),
          FIND THE FIRST CASE
          IF FOUND
               REMOVE REQ FROM UNSATISFIED.REQUESTS
               FILE REQ IN SATISFIED.REQUESTS(LEG36)
               LET SEATS.AVAILABLE(SABRELINER)=
                    SEATS.AVAILABLE(SABRELINER)-PAX.LOAD(REQ)
               JUMP BACK
          ELSE
ALWAYS
     GO TO 'RL38'
     ELSE
```

```
CALL REFUEL GIVEN L.DESTINATION(L.ITINERARY(SABRELINER)),R.ORIGIN(BSTR3),
     DIST2,GS2,TC2,LAT2,LONG2,0,0
FOR EACH RF.BASE IN REF.BASE.FILE, WITH TOT.TIME(RF.BASE) LE TIME2+.5,
     FIND THE FIRST CASE
CREATE A LEG CALLED LEG35
LET L.ORIGIN(LEG35)=L.DESTINATION(L.ITINERARY(SABRELINER))
LET L.DESTINATION(LEG35)=RF.ICAO(RF.BASE)
LET DEPARTURE.TIME(LEG35)=ARRIVAL.TIME(L.ITINERARY(SABRELINER))+1
LET ENROUTE.TIME(LEG35)=IN.TIME(RF.BASE)
LET ARRIVAL.TIME(LEG35)=DEPARTURE.TIME(LEG35)+ENROUTE.TIME(LEG35)
LET FUEL.CONSUMED(LEG35)=IN.FUEL(RF.BASE)
FILE LEG35 IN ITINERARY(SABRELINER)
CREATE A LEG CALLED LEG36
LET L.ORIGIN(LEG36)=RF.ICAO(RF.BASE)
LET L.DESTINATION(LEG36)=R.ORIGIN(BSTR3)
LET DEPARTURE.TIME(LEG36)=ARRIVAL.TIME(LEG35)+1
LET ENROUTE.TIME(LEG36)=OUT.TIME(RF.BASE)
LET ARRIVAL.TIME(LEG36)=DEPARTURE.TIME(LEG36)+ENROUTE.TIME(LEG36)
LET FUEL.CONSUMED(LEG36)=OUT.FUEL(RF.BASE)
FILE LEG36 IN ITINERARY(SABRELINER)
     LET PAXTHRU=0
     LET PAXIN=0
     LET PAXOUT=0
     LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)
     LET INSEATS=SEATS.AVAILABLE(SABRELINER)
     LET OUTSEATS=SEATS.AVAILABLE(SABRELINER)
''SEARCH FOR ADDITIONAL PASSENGERS
HERE
IF THRUSEATS GT 0 OR OUTSEATS GT 0 OR INSEATS GT 0
     FOR EACH REQ IN UNSATISFIED.REQUESTS,WITH
          (R.ORIGIN(REQ) EQ L.ORIGIN(LEG35) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG36) AND
          NET.TIME(REQ) LE DEPARTURE.TIME(LEG35) AND
          NLT.TIME(REQ) GE ARRIVAL.TIME(LEG36) AND
          PAX.LOAD(REQ) LE THRUSEATS) OR
          (R.ORIGIN(REQ) EQ L.ORIGIN(LEG35) AND
```

```
R.DESTINATION(REQ) EQ L.DESTINATION(LEG35) AND
NET.TIME(REQ) LE DEPARTURE.TIME(LEG35) AND
NLT.TIME(REQ) GE ARRIVAL.TIME(LEG35) AND
PAX.LOAD(REQ) LE INSEATS) OR
(R.ORIGIN(REQ) EQ L.ORIGIN(LEG36) AND
R.DESTINATION(REQ) EQ L.DESTINATION(LEG36) AND
NET.TIME(REQ) LE DEPARTURE.TIME(LEG36) AND
NLT.TIME(REQ) GE ARRIVAL.TIME(LEG36) AND
PAX.LOAD(REQ) LE OUTSEATS)
FIND THE FIRST CASE
IF FOUND
     REMOVE REQ FROM UNSATISFIED.REQUESTS
     IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG35) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG36)
          LET PAXTHRU=PAXTHRU+PAX.LOAD(REQ)
          LET THRUSEATS=THRUSEATS-PAX.LOAD(REQ)
          LET INSEATS=INSEATS-PAX.LOAD(REQ)
          LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ)
          FILE REQ IN SATISFIED.REQUESTS(LEG35)
     ALWAYS
     IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG35) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG35)
          LET PAXIN=PAXIN+PAX.LOAD(REQ)
          LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
               MAX.F(PAXIN,PAXOUT)
          LET INSEATS=INSEATS-PAX.LOAD(REQ)
          FILE REQ IN SATISFIED.REQUESTS(LEG35)
     ALWAYS
     IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG36) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG36)
          LET PAXOUT=PAXOUT+PAX.LOAD(REQ)
          LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
               MAX.F(PAXIN,PAXOUT)
          LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ)
          FILE REQ IN SATISFIED.REQUESTS(LEG36)
     ALWAYS
```

```
          JUMP BACK
     ELSE
ALWAYS
ALWAYS
'RL38'
LET SEATS.AVAILABLE(SABRELINER)=5-PAX.LOAD(BSTR3)
    IF HOP1 EQ 0
        CREATE A LEG CALLED LEG38
        LET L.ORIGIN(LEG38)=R.ORIGIN(BSTR3)
        LET L.DESTINATION(LEG38)=L.DESTINATION(BSTR3)
        LET DEPARTURE.TIME(LEG38)=MAX.F(NET.TIME(BSTR3),
            ARRIVAL.TIME(L.ITINERARY(SABRELINER))+1)
        LET ENROUTE.TIME(LEG38)=TIME1
        LET ARRIVAL.TIME(LEG38)=DEPARTURE.TIME(LEG38)+ENROUTE.TIME(LEG38)
        LET FUEL.CONSUMED(LEG38)=FUEL1
        FILE BSTR3 IN SATISFIED.REQUESTS(LEG38)
        FILE LEG38 IN ITINERARY(SABRELINER)
''SEARCH FOR ADDITIONAL PASSENGERS
HERE
IF SEATS.AVAILABLE(SABRELINER) GT 0
    FOR EACH REQ IN UNSATISFIED.REQUESTS, WITH
        R.ORIGIN(REQ) EQ L.ORIGIN(LEG38) AND
        R.DESTINATION(REQ) EQ L.DESTINATION(LEG38) AND
        NET.TIME(REQ) LE DEPARTURE.TIME(LEG38) AND
        NLT.TIME(REQ) GE ARRIVAL.TIME(LEG38) AND
        PAX.LOAD(REQ) LE SEATS.AVAILABLE(SABRELINER),
        FIND THE FIRST CASE
        IF FOUND
            REMOVE REQ FROM UNSATISFIED.REQUESTS
            FILE REQ IN SATISFIED.REQUESTS(LEG38)
            LET SEATS.AVAILABLE(SABRELINER)=
                SEATS.AVAILABLE(SABRELINER)-PAX.LOAD(REQ)
            JUMP BACK
        ELSE
ALWAYS
    GO TO CONTINUE
```

```
      ELSE
CALL REFUEL GIVEN R.ORIGIN(BSTR3),R.DESTINATION(BSTR3),
      DIST1,GS1,TC1,LAT1,LONG1,0,0
FOR EACH RF.BASE IN REF.BASE.FILE, WITH TOT.TIME(RF.BASE) LE TIME1+.5,
      FIND THE FIRST CASE
CREATE A LEG CALLED LEG37
LET L.ORIGIN(LEG37)=R.ORIGIN(BSTR3)
LET L.DESTINATION(LEG37)=RF.ICAO(RF.BASE)
LET DEPARTURE.TIME(LEG37)=MAX.F(NET.TIME(BSTR3),
      ARRIVAL.TIME(L.ITINERARY(SABRELINER))+1)
LET ENROUTE.TIME(LEG37)=IN.TIME(RF.BASE)
LET ARRIVAL.TIME(LEG37)=DEPARTURE.TIME(LEG37)+ENROUTE.TIME(LEG37)
LET FUEL.CONSUMED(LEG37)=IN.FUEL(RF.BASE)
FILE BSTR3 IN SATISFIED.REQUESTS(LEG37)
FILE LEG37 IN ITINERARY(SABRELINER)
CREATE A LEG CALLED LEG38
LET L.ORIGIN(LEG38)=RF.ICAO(RF.BASE)
LET L.DESTINATION(LEG38)=L.DESTINATION(BSTR3)
LET DEPARTURE.TIME(LEG38)=ARRIVAL.TIME(LEG37)+1
LET ENROUTE.TIME(LEG38)=OUT.TIME(RF.BASE)
LET ARRIVAL.TIME(LEG38)=DEPARTURE.TIME(LEG38)+ENROUTE.TIME(LEG38)
LET FUEL.CONSUMED(LEG38)=OUT.FUEL(RF.BASE)
FILE LEG38 IN ITINERARY(SABRELINER)
IF SEATS.AVAILABLE(SABRELINER) GT 0
      LET PAXTHRU=0
      LET PAXIN=0
      LET PAXOUT=0
      LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)
      LET INSEATS=SEATS.AVAILABLE(SABRELINER)
      LET OUTSEATS=SEATS.AVAILABLE(SABRELINER)
''SEARCH FOR ADDITIONAL PASSENGERS
HERE
IF THRUSEATS GT 0 OR OUTSEATS GT 0 OR INSEATS GT 0
      FOR EACH REQ IN UNSATISFIED.REQUESTS,WITH
            (R.ORIGIN(REQ) EQ L.ORIGIN(LEG37) AND
            R.DESTINATION(REQ) EQ L.DESTINATION(LEG38) AND
```

```
NET.TIME(REQ) LE DEPARTURE.TIME(LEG37) AND
NLT.TIME(REQ) GE ARRIVAL.TIME(LEG38) AND
PAX.LOAD(REQ) LE THRUSEATS) OR
(R.ORIGIN(REQ) EQ L.ORIGIN(LEG37) AND
R.DESTINATION(REQ) EQ L.DESTINATION(LEG37) AND
NET.TIME(REQ) LE DEPARTURE.TIME(LEG37) AND
NLT.TIME(REQ) GE ARRIVAL.TIME(LEG37) AND
PAX.LOAD(REQ) LE INSEATS) OR
(R.ORIGIN(REQ) EQ L.ORIGIN(LEG38) AND
R.DESTINATION(REQ) EQ L.DESTINATION(LEG38) AND
NET.TIME(REQ) LE DEPARTURE.TIME(LEG38) AND
NLT.TIME(REQ) GE ARRIVAL.TIME(LEG38) AND
PAX.LOAD(REQ) LE OUTSEATS)
FIND THE FIRST CASE
IF FOUND
     REMOVE REQ FROM UNSATISFIED.REQUESTS
     IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG37) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG38)
          LET PAXTHRU=PAXTHRU+PAX.LOAD(REQ)
          LET THRUSEATS=THRUSEATS-PAX.LOAD(REQ)
          LET INSEATS=INSEATS-PAX.LOAD(REQ)
          LET OUTSEATS=OUTSEATS-PAX,LOAD(REQ)
          FILE REQ IN SATISFIED.REQUESTS(LEG37)
     ALWAYS
     IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG37) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG37)
          LET PAXIN=PAXIN+PAX.LOAD(REQ)
          LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
               MAX.F(PAXIN,PAXOUT)
          LET INSEATS=INSEATS-PAX.LOAD(REQ)
          FILE REQ IN SATISFIED.REQUESTS(LEG37)
     ALWAYS
     IF R.ORIGIN(REQ) EQ L.ORIGIN(LEG38) AND
          R.DESTINATION(REQ) EQ L.DESTINATION(LEG38)
          LET PAXOUT=PAXOUT+PAX.LOAD(REQ)
          LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)-PAXTHRU-
```

```
                        MAX.F(PAXIN,PAXOUT)
                    LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ)
                    FILE REQ IN SATISFIED.REQUESTS(LEG38)
                ALWAYS
                JUMP BACK
            ELSE
ALWAYS
ALWAYS
GO TO CONTINUE
'RCONTINUE'
''NO OTHER REQUESTS CAN BE SATISFIED IN THE REMAINING CREW DAY;
''FLY DIRECTLY TO TERMINATION POINT.
LET DUTY.DAY(SABRELINER)=ARRIVAL.TIME(L.ITINERARY(SABRELINER))+1-
    CREW.DUTY.START(SABRELINER)
LET SEATS.AVAILABLE(SABRELINER)=5
IF L.DESTINATION(L.ITINERARY(SABRELINER)) NE HOME.STATION(SABRELINER)
        GO TO ROP1
    ELSE
        GO TO TERM
'ROP1'
        CALL LEG.DATA GIVEN L.DESTINATION(L.ITINERARY(SABRELINER)),
            HOME.STATION(SABRELINER) YIELDING DIST3,HOP3,HOMETIME,
                HOMEFUEL
CREATE A LEG CALLED LEG8
LET L.ORIGIN(LEG8)=L.DESTINATION(L.ITINERARY(SABRELINER))
LET L.DESTINATION(LEG8)=HOME.STATION(SABRELINER)
LET DEPARTURE.TIME(LEG8)=ARRIVAL.TIME(L.ITINERARY(SABRELINER))+1
LET ENROUTE.TIME(LEG8)=HOMETIME
LET ARRIVAL.TIME(LEG8)=DEPARTURE.TIME(LEG8)+ENROUTE.TIME(LEG8)
LET FUEL.CONSUMED(LEG8)=HOMEFUEL
FILE LEG8 IN ITINERARY(SABRELINER)
LET DUTY.DAY(SABRELINER)=ARRIVAL.TIME(LEG8)-CREW.DUTY.START(SABRELINER)
GO TO TERM
'TERM'
LET T39S.AVAILABLE=T39S.AVAILABLE-1
GO TO SKED
```

```
'INFEASIBLE'
PRINT 1 LINE WITH T39S.AVAILABLE, N.UNSATISFIED.REQUESTS THUS
INFEASIBLE:  ** T-39S ARE NOT SCHEDULED  *** REQUESTS ARE NOT SATISFIED
RETURN
END ''OF T39.SCHEDULER
```

```
ROUTINE LEG.DATA GIVEN ORIGIN,DESTINATION YIELDING LEG.DIST,LEG.STOPS,

     LEG.TIME,LEG.FUEL,GROUND.SPEED,TRUE.COURSE,LAT1,LONG1
''THIS ROUTINE ASSUMES A ROUND EARTH AND USES RELATIONSHIPS FROM
''SPHERICAL TRIGONOMETRY TO ESTIMATE SURFACE DISTANCES
''LEG.DIST IS THE APPROXIMATE DISTANCE (IN NAUTICAL MILES) BETWEEN THE
''ORIGIN AND THE DESTINATION
''LEG.STOPS IS THE MINIMUM NUMBER OF REFUELING STOPS REQUIRED BETWEEN
''THE ORIGIN AND THE DESTINATION BASED ON A MAXIMUM LEG LENGTH OF 3+15
''LEG.TIME IS THE APPROXIMATE FLYING BETWEEN THE ORIGIN AND THE
''DESTINATION BASED ON AN AVERAGE TRUE AIRSPEED OF 400 KNOTS, A FIXED
''TIME FOR CLIMB, DESCENT, APPROACH, AND LANDING, SEASONAL WIND
''ESTIMATES PROVIDED BY MAC/DOOF, AND A REFUELING GROUND TIME OF 1+00
''LEG.FUEL IS THE APPROXIMATE FUEL CONSUMPTION BETWEEN THE ORIGIN AND
''THE DESTINATION BASED ON AN AVERAGE FUEL CONSUMPTION OF 1900 POUNDS
''PER HOUR OF FLYING TIME
''GROUND SPEED IS THE APPROXIMATE AVERAGE GROUND SPEED BETWEEN
''THE ORIGIN AND THE DESTINATION BASED ON AN AVERAGE TRUE AIRSPEED OF
''400 KNOTS AND SEASONAL WIND ESTIMATES PROVIDED BY MAC/DOOF
''TRUE COURSE IS THE OUTBOUND GREAT CIRCLE TRUE COURSE IN RADIANS
''LAT1 IS THE LATITUDE OF THE ORIGIN IN RADIANS
''LONG1 IS THE LONGITUDE OF THE ORIGIN IN RADIANS
DEFINE
     BASE1,
     LEG.STOPS
     AS INTEGER VARIABLES
DEFINE
     ORIGIN,
     DESTINATION
     AS ALPHA VARIABLES
DEFINE TC TO MEAN TRUE.COURSE
FOR EACH BASE1 IN BASE.FILE,WITH ICAO(BASE1) EQ ORIGIN,
     FIND THE FIRST CASE
LET LAT1=LAT(BASE1)/RADIAN.C
LET LONG1=LONG(BASE1)/RADIAN.C
```

```
IF ORIGIN EQ DESTINATION
     LET LEG.DIST=0
     LET LEG.STOPS=0
     LET LEG.TIME=0
     LET LEG.FUEL=0
     LET GROUND.SPEED=0
     LET TC=0
     RETURN
ELSE
FOR EACH BASE1 IN BASE.FILE,WITH ICAO(BASE1) EQ DESTINATION,
     FIND THE FIRST CASE
LET LAT2=LAT(BASE1)/RADIAN.C
LET LONG2=LONG(BASE1)/RADIAN.C
LET POLAR.ANG=ABS.F(LONG2-LONG1)
LET COS.DIST.ANG=SIN.F(LAT1)*SIN.F(LAT2)+COS.F(LAT1)*COS.F(LAT2)*
     COS.F(POLAR.ANG)
LET DIST.ANG=ARCCOS.F(COS.DIST.ANG)
LET LEG.DIST=DIST.ANG*RADIAN.C*60.
LET TC=ARCCOS.F((SIN.F(LAT2)-SIN.F(LAT1)*COS.F(DIST.ANG))/
     (COS.F(LAT1)*SIN.F(DIST.ANG)))
LET WD=90./RADIAN.C
IF LONG2>LONG1
     LET TC=360./RADIAN.C-TC
ALWAYS
IF DAY<120 OR DAY >273
     LET WV=65.
ELSE
     LET WV=25.
ALWAYS
LET TAS=400.
LET GROUND.SPEED=WV*COS.F(WD-TC)+SQRT.F((WV*COS.F(WD-TC))**2-WV**2
     +TAS**2)
LET CRUISE.TIME=LEG.DIST/GROUND.SPEED
LET LEG.STOPS=TRUNC.F(CRUISE.TIME/2.9375)
LET FLY.TIME=CRUISE.TIME+(LEG.STOPS+1)*.3125
LET LEG.TIME=FLY.TIME+LEG.STOPS
```

```
LET LEG.FUEL=FLY.TIME*1900
RETURN
END ''OF LEG.DATA
```

```
ROUTINE REFUEL GIVEN ORIGIN,DESTINATION,DISTANCE,GROUND.SPEED,

     TRUE.COURSE,LAT1,LONG1
''THIS ROUTINE IDENTIFIES A FEASIBLE SET OF REFUELING BASES BETWEEN
''THE ORIGIN AND THE DESTINATION
''DISTANCE, GROUND.SPEED, TRUE.COURSE, LAT1, AND LONG1 ARE OUTPUTS
''FROM LEG.DATA
DEFINE
     ORIGIN,
     DESTINATION
     AS ALPHA VARIABLES
DEFINE
     THRUSEATS,
     INSEATS,
     OUTSEATS,
     PAXTHRU,
     PAXIN,
     PAXOUT,
     REQ1,
     BASE1,
     I,
     K
     AS INTEGER VARIABLES
DEFINE TC TO MEAN TRUE.COURSE
DEFINE DIST TO MEAN DISTANCE
DEFINE GS TO MEAN GROUND.SPEED
FOR EACH RF.BASE IN REF.BASE.FILE
DO
     REMOVE RF.BASE FROM REF.BASE.FILE
     DESTROY RF.BASE
LOOP
''ESTABLISH THE SEARCH REGION
''INITIALIZE
LET TOPLAT=0
LET BOTLAT=PI.C/2
```

```
LET WLONG=0
LET ELONG=PI.C
LET X2=GS*2.9375
LET X1=DIST-X2
LET DRDIST=450
CALL POSITION GIVEN X1,TC,LAT1,LONG1
      YIELDING X1LAT,X1LONG
CALL POSITION GIVEN X2,TC,LAT1,LONG1
      YIELDING X2LAT,X2LONG
FOR I=1 TO 2
DO
      IF I=1
            LET XILAT=X1LAT
            LET XILONG=X1LONG
      ELSE
            LET XILAT=X2LAT
            LET XILONG=X2LONG
      ALWAYS
      FOR K=1 TO 2
      DO
            IF K=1
                  LET PTC=TC+PI.C/2
            ELSE
                  LET PTC=TC-PI.C/2
            ALWAYS
            CALL POSITION GIVEN DRDIST,PTC,XILAT,XILONG
                  YIELDING UILAT,UILONG
            IF TOPLAT LT UILAT
                  LET TOPLAT=UILAT
            ALWAYS
            IF BOTLAT GT UILAT
                  LET BOTLAT=UILAT
            ALWAYS
            IF WLONG LT UILONG
                  LET WLONG=UILONG
            ALWAYS
```

```
        IF ELONG GT UILONG
            LET ELONG=UILONG
        ALWAYS
    LOOP
LOOP
LET TOPLAT=TOPLAT*RADIAN.C
LET BOTLAT=BOTLAT*RADIAN.C
LET WLONG=WLONG*RADIAN.C
LET ELONG=ELONG*RADIAN.C
FOR EACH BASE1 IN BASE.FILE,WITH
    LAT(BASE1) LE TOPLAT AND
    LAT(BASE1) GE BOTLAT AND
    LONG(BASE1) LE WLONG AND
    LONG(BASE1) GE ELONG AND
    ICAO(BASE1) NE ORIGIN AND
    ICAO(BASE1) NE DESTINATION
DO
    CALL LEG.DATA GIVEN ORIGIN,ICAO(BASE1)
        YIELDING X1,I,TIME1,FUEL1
    IF I=0
        CALL LEG.DATA GIVEN ICAO(BASE1),DESTINATION
            YIELDING X2,K,TIME2,FUEL2
        IF K=0
            CREATE AN RF.BASE
            LET RF.ICAO(RF.BASE)=ICAO(BASE1)
            LET IN.TIME(RF.BASE)=TIME1
            LET IN.FUEL(RF.BASE)=FUEL1
            LET OUT.TIME(RF.BASE)=TIME2
            LET OUT.FUEL(RF.BASE)=FUEL2
            LET TOT.TIME(RF.BASE)=TIME1+TIME2+1
            IF SEATS.AVAILABLE(SABRELINER) GT 0
                LET PAXTHRU=0
                LET PAXIN=0
                LET PAXOUT=0
                LET THRUSEATS=SEATS.AVAILABLE(SABRELINER)
                LET INSEATS=SEATS.AVAILABLE(SABRELINER)
```

```
LET OUTSEATS=SEATS.AVAILABLE(SABRELINER)
IF N.ITINERARY(SABRELINER) EQ 0
     LFT TO.TIME=CREW.DUTY.START(SABRELINER)+2
     LET RF.TO.TIME=TO.TIME+TIME1+1
ELSE
     LET TO.TIME=
          ARRIVAL.TIME(L.ITINERARY(SABRELINER))+1
     LET RF.TO.TIME=TO.TIME+TIME1+1
ALWAYS
FOR EACH REQ1 IN UNSATISFIED.REQUESTS,WITH
     (R.ORIGIN(REQ1) EQ ORIGIN AND
     R.DESTINATION(REQ1) EQ DESTINATION AND
     NET.TIME(REQ1) LE TO.TIME AND
     NLT.TIME(REQ1) GE RF.TO.TIME+TIME2 AND
     PAX.LOAD(REQ1) LE SEATS.AVAILABLE(SABRELINER))
     OR
     (R.ORIGIN(REQ1) EQ ORIGIN AND
     R.DESTINATION(REQ1) EQ ICAO(BASE1) AND
     NET.TIME(REQ1) LE TO.TIME AND
     NLT.TIME(REQ1) GE TO.TIME+TIME1 AND
     PAX.LOAD(REQ1) LE SEATS.AVAILABLE(SABRELINER))
     OR
     (R.ORIGIN(REQ1) EQ ICAO(BASE1) AND
     R.DESTINATION(REQ1) EQ DESTINATION AND
     NET.TIME(REQ1) LE RF.TO.TIME AND
     NLT.TIME(REQ1) GE RF.TO.TIME+TIME2 AND
     PAX.LOAD(REQ1) LE SEATS.AVAILABLE(SABRELINER))
     DO
          IF R.ORIGIN(REQ1) EQ ORIGIN AND
               R.DESTINATION(REQ1) EQ DESTINATION
               IF PAX.LOAD(REQ1) GT THRUSEATS
                    GO TO EXIT
               ELSE
               LET PAXTHRU=PAXTHRU+PAX.LOAD(REQ1)
               LET THRUSEATS=THRUSEATS-
                    PAX.LOAD(REQ1)
```

```
                                                LET INSEATS=INSEATS-PAX.LOAD(REQ1)
                                                LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ1)
                                                IF PAX.PRIORITY(REQ1) LE 3
                                                     LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+(10**(4-PAX.PRIORITY(REQ1)))+PAX.LOAD(REQ1)*(1+1/
     (PAX.PRIORITY(REQ1)+9))
                                                ELSE
                                                     LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+PAX.LOAD(REQ1)*(1+1/(PAX.PRIORITY(REQ1)+9))
                                                ALWAYS
                                           ALWAYS
                                           IF R.ORIGIN(REQ1) EQ ORIGIN AND
                                                R.DESTINATION(REQ1) EQ ICAO(BASE1)
                                                IF PAX.LOAD(REQ1) GT INSEATS
                                                     GO TO EXIT
                                                ELSE
                                                LET PAXIN=PAXIN+PAX.LOAD(REQ1)
                                                LET INSEATS=INSEATS-PAX.LOAD(REQ1)
                                                LET THRUSEATS=
                                                     SEATS.AVAILABLE(SABRELINER)-
                                                     PAXTHRU-MAX.F(PAXIN,PAXOUT)
                                                IF PAX.PRIORITY(REQ1) LE 3
                                                     LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+(10**(4-PAX.PRIORITY(REQ1)))+PAX.LOAD(REQ1)*(1+1/
     (PAX.PRIORITY(REQ1)+9))
                                                ELSE
                                                     LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+PAX.LOAD(REQ1)*(1+1/(PAX.PRIORITY(REQ1)+9))
                                                ALWAYS
                                           ALWAYS
                                           IF R.ORIGIN(REQ1) EQ ICAO(BASE1) AND
                                                R.DESTINATION(REQ1) EQ DESTINATION
                                                IF PAX.LOAD(REQ1) GT OUTSEATS
                                                     GO TO EXIT
                                                ELSE
                                                LET PAXOUT=PAXOUT+PAX.LOAD(REQ1)
```

```
                                    LET OUTSEATS=OUTSEATS-PAX.LOAD(REQ1)
                                    LET THRUSEATS=
                                        SEATS.AVAILABLE(SABRELINER)-
                                        PAXTHRU-MAX.F(PAXIN,PAXOUT)
                                    IF PAX.PRIORITY(REQ1) LE 3
                                        LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+(10**(4-PAX.PRIORITY(REQ1)))+PAX.LOAD(REQ1)*(1+1/
    (PAX.PRIORITY(REQ1)+9))
                                    ELSE
                                        LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+PAX.LOAD(REQ1)*(1+1/(PAX.PRIORITY(REQ1)+9))
                                    ALWAYS
                                ALWAYS
                                'EXIT'
                            LOOP
                    ALWAYS
                    FILE RF.BASE IN REF.BASE.FILE
                ALWAYS
            ALWAYS
        LOOP
        RETURN
        END ''OF REFUEL
```

```
ROUTINE POSITION GIVEN DRDIST,PTC,XILAT,XILONG


     YIELDING UILAT,UILONG
''THIS ROUTINE USES RELATIONSHIPS FROM SPHERICAL TRIGONOMETRY TO
''DETERMINE COORDINATES OF POINTS USED BY REFUEL TO ESTABLISH THE
''SEARCH REGION
LET UILAT=XILAT+COS.F(PTC)*DRDIST/(60*RADIAN.C)
IF ABS.F(SIN.F(PTC))LE .999
     LET UILONG=XILONG+TAN.F(PTC)*LOG.E.F(TAN.F(PI.C/4
          +XILAT/2)/TAN.F(PI.C/4+UILAT/2))
ELSE
     LET UILONG=XILONG-SIN.F(PTC)*DRDIST/(60*RADIAN.C*COS.F(XILAT))
ALWAYS
RETURN
END ''OF POSITION
```

```
ROUTINE INTERPLANE


DEFINE
     CK.BASE,
     CK.HOME
     AS ALPHA VARIABLES
DEFINE
     REQ1,REQ2,REQ3,PEQ4,REQ5,
     REQA,
     BASE1,BASE2,BASE3,BASE4,BASE5,
     SOURCE,
     INSEATS,
     OUTSEATS,
     SAB1,SAB2
     AS INTEGER VARIABLES
IF N.REF.BASE.FILE GT 0
     FOR EACH RF.BASE IN REF.BASE.FILE
     DO
          REMOVE RF.BASE FPOM REF.BASE.FILE
          DESTROY RF.BASE
     LOOP
ALWAYS
LET REQ1=IPRQPT
REMOVE REQ1 FROM UNSATISFIED.REQUESTS
IF T39S.AVAILABLE LT 2 OR PAX.PRIORITY(REQ1) GT 3
     LET PAX.PRIORITY(REQ1)=PAX.PRIORITY(REQ1)+20
     FILE REQ1 IN UNSATISFIED.REQUESTS
     GO TO DONE
ELSE
FOR EACH BASE1 IN BASE.FILE,WITH ICAO(BASE1) EQ R.ORIGIN(REQ1),
     FIND THE FIRST CASE
     IF NONE
          PRINT 1 LINE THUS
INTERPLANE TERMINATE 1
          STOP
```

```
     ELSE
FOR EACH BASE2 IN BASE.FILE,WITH ICAO(BASE2)  EQ R.DESTINATION(REQ1),
     FIND THE FIRST CASE
     IF NONE
          PRINT 1 LINE THUS
INTERPLANE TERMINATE 2
          STOP
     ELSE
LET BOTLAT=(LAT(BASE1)+LAT(BASE2))/2-7.5
LET TOPLAT=BOTLAT+15.0
LET ELONG=(LONG(BASE1)+LONG(BASE2))/2-9.5
LET WLONG=ELONG+19.0
FOR EACH BASE3 IN BASE.FILE,WITH N.DET(BASE3) GT 0
DO
     FOR EACH SAB1 IN DET(BASE3),WITH
          DUTY.DAY(SAB1)  EQ 0 AND
          HOME.STATION(SAB1) NE ICAO(BASE3),
          FIND THE FIRST CASE
          IF FOUND
               LET BOTLAT=MIN.F(BOTLAT,LAT(BASE1),LAT(BASE2))
               LET TOPLAT=MAX.F(TOPLAT,LAT(BASE1),LAT(BASE2))
               LET ELONG=MIN.F(ELONG,LONG(BASE1),LONG(BASE2))
               LET WLONG=MAX.F(WLONG,LONG(BASE1),LONG(BASE2))
               JUMP AHEAD
          ELSE
LOOP
HERE
FOR EACH BASE3 IN BASE.FILE,WITH
     LAT(BASE3) GE BOTLAT AND
     LAT(BASE3) LE TOPLAT AND
     LONG(BASE3) GE ELONG AND
     LONG(BASE3) LE WLONG AND
     ICAO(BASE3) NE R.ORIGIN(REQ1) AND
     ICAO(BASE3) NE R.DESTINATION(REQ1)
DO
     CALL LEG.DATA GIVEN R.ORIGIN(REQ1),ICAO(BASE3)
```

```
        YIELDING X,X,TIME1
FOR EACH BASE1 IN BASE.FILE,WITH ICAO(BASE1) EQ R.ORIGIN(REQ1)
DO
    IF N.DET(BASE1) EQ 0
        JUMP AHEAD
    ELSE
    IF NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE1)-2 GE 6 AND
        NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE1)-2 LE 10
        LET MAX=14
    ELSE
        LET MAX=12
    ALWAYS
    FOR EACH SAB1 IN DET(BASE1),WITH DUTY.DAY(SAB1) EQ 0,
        FIND THE FIRST CASE
        IF NONE
            JUMP AHEAD
        ELSE
        IF ICAO(BASE3) EQ HOME.STATION(SAB1)
            IF TIME1 LE MAX-2
                REMOVE SAB1 FROM DET(BASE1)
                LET SOURCE=1
                GO TO 1CONTINUE
            ELSE
        ELSE
                CALL LEG.DATA GIVEN ICAO(BASE3),HOME.STATION(SAB1)
                    YIELDING X,X,TIME2
                IF TIME1+TIME2 LE MAX-3
                    REMOVE SAB1 FROM DET(BASE1)
                    LET SOURCE=1
                    GO TO 1CONTINUE
                ELSE
        ALWAYS
    HERE
LOOP
    LET CK.PAD=0
    FOR EACH BASE4 IN BASE.FILE,WITH
```

```
          N.DET(BASE4) GT 0 AND
          ICAO(BASE4) NE R.ORIGIN(REQ1)
      DO
          FOR EACH SAB1 IN DET(BASE4),WITH DUTY.DAY(SAB1) EQ 0,
              FIND THE FIRST CASE
              IF NONE
                  JUMP AHEAD
              ELSE
              CALL LEG.DATA GIVEN ICAO(BASE4),R.ORIGIN(REQ1)
                  YIELDING X,X,TIME2
              IF NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE4)-TIME2-3 GE 6 AND
                  NET.TIME(REQ1)-LCL.TIME.CHANGE(BASE4)-TIME2-3 LE 10
                  LET MAX=14
              ELSE
                  LET MAX=12
              ALWAYS
              IF ICAO(BASE3) EQ HOME.STATION(SAB1)
                  LET PAD=MAX-TIME1-TIME2-3
                  IF PAD GT CK.PAD
                      LET CK.PAD=PAD
                      LET CK.BASE=ICAO(BASE4)
                      LET CK.HOME=HOME.STATION(SAB1)
                  ALWAYS
              ELSE
                  CALL LEG.DATA GIVEN ICAO(BASE3),
                      HOME.STATION(SAB1)
                      YIELDING X,X,TIME3
                      LET PAD=MAX-TIME1-TIME2-TIME3-4
                      IF PAD GT CK.PAD
                          LET CK.PAD=PAD
                          LET CK.BASE=ICAO(BASE4)
                          LET CK.HOME=HOME.STATION(SAB1)
                      ALWAYS
              ALWAYS
              HERE
  LOOP
```

```
IF CK.PAD EQ 0
     GO TO PUNT
ELSE
     FOR EACH BASE4 IN BASE.FILE,WITH ICAO(BASE4) EQ CK.BASE
     DO
          FOR EACH SAB1 IN DET(BASE4),WITH
               DUTY.DAY(SAB1) EQ 0 AND
               HOME.STATION(SAB1) EQ CK.HOME,
               FIND THE FIPST CASE
               IF FOUND
                    REMOVE SAB1 FROM DET(BASE4)
                    LET SOURCE=4
                    GO TO 1CONTINUE
               ELSE
     LOOP
'1CONTINUE'
''FIRST HALF OF MISSION IS FEASIBLE WITHIN MAXIMUM CREW DUTY DAY
     CALL LEG.DATA GIVEN ICAO(BASE3),R.DESTINATION(REQ1)
          YIELDING X,X,TIME5
     IF N.DET(BASE3) EQ 0
          JUMP AHEAD
     ELSE
     IF NET.TIME(REQ1)+TIME1-LCL.TIME.CHANGE(BASE3)-1 GE 6 AND
          NET.TIME(REQ1)+TIME1-LCL.TIME.CHANGE(BASE3)-1 LE 10
          LET MAX=14
     ELSE
          LET MAX=12
     ALWAYS
     FOR EACH SAB2 IN DET(BASE3),WITH DUTY.DAY(SAB2) EQ 0,
          FIND THE FIRST CASE
          IF NONE
               JUMP AHEAD
          ELSE
          IF R.DESTINATION(REQ1) EQ HOME.STATION(SAB2)
               IF TIME5 LE MAX-2
                    IF SOURCE EQ 1
```

```
                    FILE SAB1 IN DET(BASE1)
               ELSE
                    FILE SAB1 IN DET(BASE4)
               ALWAYS
               GO TO 2CONTINUE
          ELSE
     ELSE
               CALL LEG.DATA GIVEN R.DESTINATION(REQ1),
                    HOME.STATION(SAB2)
                    YIELDING X,X,TIME6
               IF TIME5+TIME6 LE MAX-3
                    IF SOURCE EQ 1
                         FILE SAB1 IN DET(BASE1)
                    ELSE
                         FILE SAB1 IN DET(BASE4)
                    ALWAYS
                    GO TO 2CONTINUE
               ELSE
     ALWAYS
HERE
LET CK.PAD=0
FOR EACH BASE5 IN BASE.FILE,WITH
     N.DET(BASE5) GT 0 AND
     ICAO(BASE5) NE ICAO(BASE3)
DO
     FOR EACH SAB2 IN DET(BASE5),WITH DUTY.DAY(SAB2) EQ 0,
          FIND THE FIRST CASE
          IF NONE
               JUMP AHEAD
          ELSE
          CALL LEG.DATA GIVEN ICAO(BASE5),ICAO(BASE3)
               YIELDING X,X,TIME6
          IF NET.TIME(REQ1)+TIME1-LCL.TIME.CHANGE(BASE5)-
               TIME6-1 GE 6 AND
               NET.TIME(REQ1)+TIME1-LCL.TIME.CHANGE(BASE5)-
               TIME6-1 LE 10
```

```
                    LET MAX=14
                ELSE
                    LET MAX=12
                ALWAYS
                IF R.DESTINATION(REQ1) EQ HOME.STATION(SAB2)
                    LET PAD=MAX-TIME5-TIME6-3
                    IF PAD GT CK.PAD
                        LET CK.PAD=PAD
                        LET CK.BASE=ICAO(BASE5)
                        LET CK.HOME=HOME.STATION(SAB2)
                    ALWAYS
                ELSE
                    CALL LEG.DATA GIVEN R.DESTINATION(REQ1),
                        HOME.STATION(SAB2)
                        YIELDING X,X,TIME7
                        LET PAD=MAX-TIME5-TIME6-TIME7-4
                        IF PAD GT CK.PAD
                            LET CK.PAD=PAD
                            LET CK.BASE=ICAO(BASE5)
                            LET CK.HOME=HOME.STATION(SAB2)
                        ALWAYS
                ALWAYS
                HERE
        LOOP
    IF CK.PAD EQ 0
        IF SOURCE EQ 1
            FILE SAB1 IN DET(BASE1)
        ELSE
            FILE SAB1 IN DET(BASE4)
        ALWAYS
        GO TO PUNT
    ELSE
        FOR EACH BASE5 IN BASE.FILE,WITH ICAO(BASE5) EQ CK.BASE
        DO
            FOR EACH SAB2 IN DET(BASE5),WITH
                DUTY.DAY(SAB2) EQ 0 AND
```

```
                HOME.STATION(SAB2) EQ CK.HOME,
                FIND THE FIRST CASE
                IF FOUND
                     IF SOURCE EQ 1
                          FILE SAB1 IN DET(BASE1)
                     ELSE
                          FILE SAB1 IN DET(BASE4)
                     ALWAYS
                     GO TO 2CONTINUE
                ELSE
     LOOP
'2CONTINUE'
''SECOND HALF OF MISSION IS FEASIBLE WITHIN MAXIMUM CREW DUTY DAY
CREATE AN RF.BASE
LET RF.ICAO(RF.BASE)=ICAO(BASE3)
LET IN.TIME(RF.BASE)=TIME1
LET OUT.TIME(RF.BASE)=TIME5
LET TOT.TIME(RF.BASE)=IN.TIME(RF.BASE)+OUT.TIME(RF.BASE)+1
IF PAX.LOAD(REQ1) LT 5
     LET PAXIN=0
     LET PAXOUT=0
     LET INSEATS=5-PAX.LOAD(REQ1)
     LET OUTSEATS=5-PAX.LOAD(REQ1)
     LET TO.TIME=NET.TIME(REQ1)
     LET RF.TO.TIME=TO.TIME+TIME1+1
     FOR EACH REQA IN UNSATISFIED.REQUESTS,WITH
          (R.ORIGIN(REQA) EQ R.ORIGIN(REQ1) AND
          R.DESTINATION(REQA) EQ RF.ICAO(RF.BASE) AND
          NET.TIME(REQA) LE TO.TIME AND
          NLT.TIME(REQA) GE TO.TIME+TIME1 AND
          PAX.LOAD(REQA) LE INSEATS)
          OR
          (R.ORIGIN(REQA) EQ RF.ICAO(RF.BASE) AND
          R.DESTINATION(REQA) EQ R.DESTINATION(REQ1) AND
          NET.TIME(REQA) LE RF.TO.TIME AND
          NLT.TIME(REQA) GE RF.TO.TIME+TIME5 AND
```

```
          PAX.LOAD(REQA) LE OUTSEATS)
          DO
               IF R.ORIGIN(REQA) EQ R.ORIGIN(REQ1) AND
                    R.DESTINATION(REQA) EQ RF.ICAO(RF.BASE)
                    IF PAX.LOAD(REQA) GT INSEATS
                         GO TO EXIT
                    ELSE
                    LET PAXIN=PAXIN+PAX.LOAD(REQA)
                    LET INSEATS=INSEATS-PAX.LOAD(REQA)
                    IF PAX.PRIORITY(REQA) LE 3
                         LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+(10**(4-PAX.PRIORITY(REQ1)))+PAX.LOAD(REQ1)*(1+1/
     (PAX.PRIORITY(REQA)+9))
                    ELSE
                         LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+PAX.LOAD(REQA)*(1+1/(PAX.PRIORITY(REQA)+9))
                    ALWAYS
               ALWAYS
               IF R.ORIGIN(REQA) EQ RF.ICAO(RF.BASE) AND
                    R.DESTINATION(REQA) EQ R.DESTINATION(REQ1)
                    IF PAX.LOAD(REQA) GT OUTSEATS
                         GO TO EXIT
                    ELSE
                    LET PAXOUT=PAXOUT+PAX.LOAD(REQA)
                    LET OUTSEATS=OUTSEATS-PAX.LOAD(REQA)
                    IF PAX.PRIORITY(REQA) LE 3
                         LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+(10**(4-PAX.PRIORITY(REQ1)))+PAX.LOAD(REQ1)*(1+1/
     (PAX.PRIORITY(REQA)+9))
                    ELSE
                         LET PAX.VALUE(RF.BASE)=
PAX.VALUE(RF.BASE)+PAX.LOAD(REQA)*(1+1/(PAX.PRIORITY(REQA)+9))
                    ALWAYS
               ALWAYS
               'EXIT'
          LOOP
```

```
ALWAYS
FILE RF.BASE IN REF.BASE.FILE
'PUNT'
LOOP
IF N.REF.BASE.FILE EQ 0
     LET PAX.PRIORITY(REQ1)=PAX.PRIORITY(REQ1)+20
     FILE REQ1 IN UNSATISFIED.REQUESTS
     PRINT 1 LINE WITH PAX.1NAME(REQ1),R.ORIGIN(REQ1),
          R.DESTINATION(REQ1) THUS
UNABLE TO INTERPLANE *****'**** FROM **** TO ****; REQUEST NOT SATISFIED.
     GO TO DONE
ELSE
REMOVE FIRST RF.BASE FROM REF.BASE.FILE
CREATE A REQ CALLED REQ2
LET R.ORIGIN(REQ2)=R.ORIGIN(REQ1)
LET R.DESTINATION(REQ2)=RF.ICAO(RF.BASE)
LET NET.TIME(REQ2)=NET.TIME(REQ1)
LET NLT.TIME(REQ2)=NET.TIME(REQ2)+IN.TIME+.5
LET PAX.LOAD(REQ2)=PAX.LOAD(REQ1)
LET PAX.PRIORITY(REQ2)=PAX.PRIORITY(REQ1)
LET DV.CODE(REQ2)=DV.CODE(REQ1)
LET PAX.1NAME(REQ2)=PAX.1NAME(REQ1)
LET PAX.2NAME(REQ2)="XXXXX"
LET PAX.RANK(REQ2)=PAX.RANK(REQ1)
FILE REQ2 IN UNSATISFIED.REQUESTS
CREATE A REQ CALLED REQ3
LET R.ORIGIN(REQ3)=RF.ICAO(RF.BASE)
LET R.DESTINATION(REQ3)=R.DESTINATION(REQ1)
LET NET.TIME(REQ3)=NLT.TIME(REQ2)+.5
LET NLT.TIME(REQ3)=NET.TIME(REQ3)+OUT.TIME+.5
LET PAX.LOAD(REQ3)=PAX.LOAD(REQ1)
LET PAX.PRIORITY(REQ3)=PAX.PRIORITY(REQ1)
LET DV.CODE(REQ3)=DV.CODE(REQ1)
LET PAX.1NAME(REQ3)=PAX.1NAME(REQ1)
LET PAX.2NAME(REQ3)="XXXXX"
LET PAX.RANK(REQ3)=PAX.RANK(REQ1)
```

```
FILE REQ3 IN UNSATISFIED.REQUESTS
DESTROY REQ CALLED REQ1
FOR EACH REQ1 IN UNSATISFIED.REQUESTS,WITH
      R.ORIGIN(REQ1) EQ R.ORIGIN(REQ2) AND
      R.DESTINATION(REQ1) EQ R.DESTINATION(REQ3)
DO
      CREATE A REQ CALLED REQ4
      LET R.ORIGIN(REQ4)=R.ORIGIN(REQ1)
      LET R.DESTINATION(REQ4)=RF.ICAO(RF.BASE)
      LET NET.TIME(REQ4)=NET.TIME(REQ1)
      LET NLT.TIME(REQ4)=NET.TIME(REQ4)+IN.TIME+.5
      LET PAX.LOAD(REQ4)=PAX.LOAD(REQ1)
      LET PAX.PRIORITY(REQ4)=PAX.PRIORITY(REQ1)+20
      LET DV.CODE(REQ4)=DV.CODE(REQ1)
      LET PAX.1NAME(REQ4)=PAX.1NAME(REQ1)
      LET PAX.2NAME(REQ4)="ZZZZZ"
      LET PAX.RANK(REQ4)=PAX.RANK(REQ1)
      FILE REQ4 IN UNSATISFIED.REQUESTS
      CREATE A REQ CALLED REQ5
      LET R.ORIGIN(REQ5)=RF.ICAO(RF.BASE)
      LET R.DESTINATION(REQ5)=R.DESTINATION(REQ1)
      LET NET.TIME(REQ5)=NLT.TIME(REQ4)+.5
      LET NLT.TIME(REQ5)=NET.TIME(REQ5)+OUT.TIME+.5
      LET PAX.LOAD(REQ5)=PAX.LOAD(REQ1)
      LET PAX.PRIORITY(REQ5)=PAX.PRIORITY(REQ1)+20
      LET DV.CODE(REQ5)=DV.CODE(REQ1)
      LET PAX.1NAME(REQ5)=PAX.1NAME(REQ1)
      LET PAX.2NAME(REQ5)="ZZZZZ"
      LET PAX.RANK(REQ5)=PAX.RANK(REQ1)
      FILE REQ5 IN UNSATISFIED.REQUESTS
LOOP
PRINT 1 LINE WITH RF.ICAO(RF.BASE) THUS
**** IS INTERPLANE BASE
'DONE'
RETURN
END ''OF INTERPLANE
```

```
ROUTINE TO PRINT.SCHEDULE


DEFINE REQX,
     DUMMY,
     LDAY,
     ZDAY,
     I,J,K
     AS INTEGER VARIABLES
DEFINE LPRANK AS AN ALPHA VARIABLE
DEFINE POINT AS AN ALPHA,1-DIMENSIONAL ARRAY
RESERVE POINT AS 20
CREATE A BASE CALLED DUMMY
PRINT 3 LINES WITH DAY THUS

AIRCRAFT SCHEDULE FOR JULIAN DAY ***

FOR EACH BASE IN BASE.FILE,WITH N.DET(BASE) GT 0,
DO
     PRINT 1 LINE THUS
################################################################################
     SKIP 1 LINE
     WRITE NAME1(BASE),NAME2(BASE),NAME3(BASE)
          AS 2 A 10,A 5
     SKIP 1 LINE
     PRINT 4 LINES THUS

                              PASSENGER LOAD DATA
DEPART TIME(Z) ARRIVE TIME(Z) #  RANKING PASSENGER   PRIORITY DESTINATION
--------------------------------------------------------------------------------
''INSURE THAT MISSIONS WILL BE LISTED IN CHRONOLOGICAL ORDER
     FOR EACH SABRELINER IN DET(BASE)
     DO
          REMOVE SABRELINER FROM DET(BASE)
          FILE SABRELINER IN DET(DUMMY)
     LOOP
```

```
    FOR EACH SABRELINER IN DET(DUMMY)
    DO
         REMOVE SABRELINER FROM DET(DUMMY)
         FILE SABRELINER IN DET(BASE)
    LOOP
    FOR EACH SABRELINER IN DET(BASE),WITH N.ITINERARY(SABRELINER) GT 0
    DO
         FOR EACH LEG IN ITINERARY(SABRELINER)
        DO
              FOR EACH REQ IN SATISFIED.REQUESTS(LEG)
                   WITH PAX.RANK EQ "OO/A"
              FIND THE FIRST CASE
                   IF FOUND
                        PRINT 3 LINES THUS
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    X NOTE: PASSENGER IS AN O-10.  MANUALLY SCHEDULE MISSION. X
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                   ELSE
         LOOP
                   ALWAYS
         LET K=0
         IF DEPARTURE.TIME(F.ITINERARY(SABRELINER))-
              CREW.DUTY.START(SABRELINER) LT 1.9167
              PRINT 4 LINES THUS

DUE TO REFUELING REQUIREMENTS, CREW DUTY START TIME AND FIRST DEPARTURE
TIME DO NOT AGREE.  CREW DUTY START TIME OR FIRST TWO DEPARTURE TIMES
MAY BE ADJUSTED.  CHECK MAXIMUM CREW DUTY DAY.
          ALWAYS
               LET OCDTIME=ARRIVAL.TIME(L.ITINERARY(SABRELINER))-
               CREW.DUTY.START(SABRELINER)-MAX.DUTY.DAY(SABRELINER)
          IF OCDTIME GT 0
               PRINT 3 LINES WITH OCDTIME THUS

CREW DUTY DAY EXCEEDS MAX DUTY DAY BY *.** HOURS.  MINOR ADJUSTMENTS
TO GROUND TIMES WILL BRING MISSION WITHIN CREW DAY LIMITATIONS.
```

```
          ALWAYS
          LET X=TRUNC.F(CREW.DUTY.START(SABRELINER))
          LET Y=CREW.DUTY.START(SABRELINER)-X
          IF INT.F(Y*60)=60
               LET X=X+1
               LET Y=0
          ALWAYS
          LET CREW.DUTY.START(SABRELINER)=(X+Y*6/10)*100
          LET X=CREW.DUTY.START(SABRELINER)-LCL.TIME.CHANGE(BASE)*100
          IF CREW.DUTY.START(SABRELINER) LT 0
               LET CREW.DUTY.START(SABRELINER)=2400+
                    CREW.DUTY.START(SABRELINER)
               LET ZDAY=DAY-1
          ELSE
               LET ZDAY=DAY
          ALWAYS
          IF X LT 0
               LET X=2400+X
               LET LDAY=DAY-1
          ELSE
               LET LDAY=DAY
          ALWAYS
          PRINT 2 LINES WITH ZDAY,CREW.DUTY.START(SABRELINER),LDAY,X THUS
CREW DUTY START   GMT DAY  GMT TIME    LOCAL DAY  LOCAL TIME
                    ***      ****          ***        ****
          FOR EACH LEG IN ITINERARY(SABRELINER)
          DO
               IF ENROUTE.TIME(LEG) GT 4 AND ENROUTE.TIME(LEG) LT 8
                    SKIP 1 LINE
                    PRINT 1 LINE THUS
THE LEG BELOW REQUIRES MANUAL SELECTION OF 1 REFUELING BASE
               ALWAYS
               IF ENROUTE.TIME(LEG) GT 8
                    SKIP 1 LINE
                    PRINT 1 LINE THUS
THE LEG BELOW REQUIRES MANUAL SELECTION OF 2 REFUELING BASES
```

```
ALWAYS
LET K=K+2
LET POINT(K-1)=L.ORIGIN(LEG)
LET POINT(K)=L.DESTINATION(LEG)
LET X=TRUNC.F(DEPARTURE.TIME(LEG))
LET Y=DEPARTURE.TIME(LEG)-X
IF INT.F(Y*60)=60
     LET X=X+1
     LET Y=0
ALWAYS
LET DEPARTUPE.TIME(LEG)=(X+Y*6/10)*100
IF DEPARTURE.TIME(LEG) GT 2400
     LET DEPARTURE.TIME(LEG)=DEPARTURE.TIME(LEG)-
          (TRUNC.F(DEPARTURE.TIME(LEG)/2400))*2400
     IF DEPARTURE.TIME(LEG) EQ 0
          LET DEPARTURE.TIME(LEG)=2400
     ALWAYS
ALWAYS
LET X=TRUNC.F(ARRIVAL.TIME(LEG))
LET Y=ARRIVAL.TIME(LEG)-X
IF INT.F(Y*60)=60
     LET X=X+1
     LET Y=0
ALWAYS
LET ARRIVAL.TIME(LEG)=(X+Y*6/10)*100
IF ARRIVAL.TIME(LEG) GT 2400
     LET ARPIVAL.TIME(LEG)=ARRIVAL.TIME(LEG)-
          (TRUNC.F(ARRIVAL.TIME(LEG)/2400))*2400
     IF ARRIVAL.TIME(LEG) EQ 0
          LET ARRIVAL.TIME(LEG)=2400
     ALWAYS
ALWAYS
WRITE L.ORIGIN(LEG),
     DEPARTURE.TIME(LEG),
     L.DESTINATION(LEG),
     ARRIVAL.TIME(LEG)
```

```
            AS /,B 2,A 4,I 7,S 3,A 4,I 7
        FOR EACH REQ IN SATISFIED.REQUESTS(LEG)
        DO
            WRITE PAX.LOAD(REQ),
                PAX.RANK(REQ),
                PAX.1NAME(REQ),
                PAX.2NAME(REQ),
                PAX.PRIORITY(REQ),
                R.DESTINATION(REQ)
                AS /,B 31,I 1,S 2,A 5,A 10,A 5,I 5,S 7,A 4
                IF R.DESTINATION(REQ) NE L.DESTINATION(LEG)
                    REMOVE REQ FROM SATISFIED.REQUESTS(LEG)
                    IF LEG NE L.ITINERARY(SABRELINER)
                    FILE REQ IN
                        SATISFIED.REQUESTS(S.ITINERARY(LEG))
                    ALWAYS
                ALWAYS
        LET LPRANK=PAX.RANK(REQ)
        LOOP
    LOOP
    IF LPRANK EQ "00/A"
        JUMP AHEAD
    ELSE
    SKIP 1 LINE
    PRINT 3 LINES THUS
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+UNSATISFIED REQUESTS THAT MAY BE COMPATIBLE WITH THIS ROUTING+
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    FOR I=1 TO N.ITINERARY(SABRELINER)*2-1 BY 2
    DO
        FOR J=I+1 TO N.ITINERARY(SABRELINER)*2 BY 2
        DO
            FOR EACH REQX IN UNSATISFIED.REQUESTS WITH R.ORIGIN(REQX)
                EQ POINT(I) AND R.DESTINATION(REQX) EQ POINT(J)
            DO
                LET W=TRUNC.F(NET.TIME(REQX))
```

```
    LET X=NET.TIME(REQX)-W
    IF INT.F(X*60)=60
        LET W=W+1
        LET X=0
    ALWAYS
    LET W=(W+X*5/10)*100
    IF W GT 2400
    LET W=W-(TRUNC.F(W/2400))*2400
        IF W EQ 0
            LET W=2400
        ALWAYS
    ALWAYS
    LET Y=TRUNC.F(NLT.TIME(REQX))
    LET Z=NLT.TIME(REQX)-Y
    IF INT.F(Z*60)=60
        LET Y=Y+1
        LET Z=0
    ALWAYS
    LET Y=(Y+Z*5/10)*100
    IF Y GT 2400
    LET Y=Y-(TRUNC.F(Y/2400))*2400
        IF Y EQ 0
            LET Y=2400
        ALWAYS
    ALWAYS
IF PAX.PRIORITY(REQX) GT 20
    LET K=PAX.PRIORITY(REQX)-20
ELSE
    LET K=PAX.PRIORITY(REQX)
ALWAYS
    WRITE R.ORIGIN(REQX),
    W,
    R.DESTINATION(REQX),
    Y,
    PAX.LOAD(REQX),
    PAX.RANK(REQX),
```

```
                              PAX.1NAME(REQX),
                              PAX.2NAME(REQX),
                         K,
                              R.DESTINATION(REQX)
                              AS /,B 2, A 4,I 7,S 3,A 4,I 7, S 4,I 1,S 2,A 5,
                                  A 10,A 5,I 5,S 7,A 4
                         LOOP
                    LOOP
               LOOP
                    HERE
          PRINT 2 LINES THUS

-------------------------------------------------------------------------------
          LOOP
     LOOP
     FOR EACH REQ IN UNSATISFIED.REQUESTS,WITH PAX.2NAME(REQ) EQ "ZZZZZ"
     DO
          REMOVE REQ FROM UNSATISFIED.REQUESTS
          DESTROY REQ
     LOOP
     FOR EACH REQ IN UNSATISFIED.REQUESTS,WITH PAX.PRIORITY(REQ) GT 20
     DO
          REMOVE REQ FROM UNSATISFIED.REQUESTS
          LET PAX.PRIORITY(REQ)=PAX.PRIORITY(REQ)-20
          FILE REQ IN PENDING.REQUESTS
     LOOP
     FOR EACH REQ IN PENDING.REQUESTS
     DO
          REMOVE REQ FROM PENDING.REQUESTS
          FILE REQ IN UNSATISFIED.REQUESTS
     LOOP
     FOR EACH REQ IN UNSATISFIED.REQUESTS,WITH PAX.2NAME(REQ) EQ "XXXXX",
          FIND THE FIRST CASE
          IF FOUND
               PRINT 7 LINES THUS
```

```
THE REQUEST BELOW WITH "XXXXX" FOLLOWING THE NAME OF THE REQUESTING
PASSENGER INDICATES THAT THE INTERPLANE ROUTINE HAS FAILED.  THE
REQUEST FOR THIS PASSENGER MUST BE REMOVED FROM THE FILE AND
SCHEDULED MANUALLY.  AFTER REMOVING ALL SUCH REQUESTS, RUN THE RROGRAM
AGAIN.

     ALWAYS
IF N.UNSATISFIED.REQUESTS GT 0
FOR EACH REQ IN UNSATISFIED.REQUESTS
DO
     REMOVE REQ FROM UNSATISFIED.REQUESTS
     FILE REQ IN PRT.UNSAT.REQ
LOOP
PRINT 5 LINES THUS

UNSUPPORTED PRIORITY 1-3 TRAVEL REQUESTS

ORGIN,  DEST    NET NET  NLT NLT  NO.   PAX      DV  NAME-FIRST         RANK/
ICAO-ID ICAO-ID DAY TIME DAY TIME PAX PRIORITY CODE PASSENGER
FOR EACH REQ IN PRT.UNSAT.REQ WITH PAX.PRIORITY(REQ) LE 3
DO
     IF NLT.TIME(REQ) GT 36.0
          REMOVE REQ FROM PRT.UNSAT.REQ
          FILE REQ IN PENDING.REQUESTS
     ELSE
     IF NLT.DATE(REQ) GT YEAR.DAYS
          LET NLT.DATE(REQ)=NLT.DATE(REQ)-YEAR.DAYS
     ALWAYS
     IF NET.DATE(REQ) GT YEAR.DAYS
          LET NET.DATE(REQ)=NET.DATE(REQ)-YEAR.DAYS
     ALWAYS
     LET X=TRUNC.F(NET.TIME(REQ))
     LET Y=NET.TIME(REQ)-X
     LET NET.TIME(REQ)=(X+Y*6/10)*100
     IF NET.TIME(REQ) GT 2400
     LET NET.TIME(REQ)=NET.TIME(REQ)-(TRUNC.F(NET.TIME(REQ)/2400))*2400
```

```
        IF NET.TIME(REQ) EQ 0
            LET NET.TIME(REQ)=2400
        ALWAYS
    ALWAYS
    LET X=TRUNC.F(NLT.TIME(REQ))
    LET Y=NLT.TIME(REQ)-X
    LET NLT.TIME(REQ)=(X+Y*6/10)*100
    IF NLT.TIME(REQ) GT 2400
    LET NLT.TIME(REQ)=NLT.TIME(REQ)-(TRUNC.F(NLT.TIME(REQ)/2400))*2400
        IF NLT.TIME(REQ) EQ 0
            LET NLT.TIME(REQ)=2400
        ALWAYS
    ALWAYS
    WRITE
        R.ORIGIN(TRAVEL.PEQUEST),
        R.DESTINATION(TRAVEL.REQUEST),
        NET.DATE(TRAVEL.PEQUEST),
        NET.TIME(TRAVEL.REQUEST),
        NLT.DATE(TRAVEL.REQUEST),
        NLT.TIME(TRAVEL.REQUEST),
        PAX.LOAD(TRAVEL.REQUEST),
        PAX.PRIORITY(TRAVEL.REQUEST),
        DV.CODE(TRAVEL.REQUEST),
        PAX.1NAME(TRAVEL.REQUEST),
        PAX.2NAME(TRAVEL.REQUEST),
        PAX.RANK(TRAVEL.REQUEST)
        AS S 1,A 8,A 7,I 3,I 5,I 4,I 5,I 4,I 5,I 7,S 3,A 10,A 7,A 4,/
    ALWAYS
LOOP
PRINT 5 LINES THUS

UNSUPPORTED PRIORITY 4-12 TRAVEL REQUESTS

ORGIN   DEST    NET NET  NLT NLT  NO.   PAX     DV  NAME-FIRST         RANK/
ICAO-ID ICAO-ID DAY TIME DAY TIME PAX PRIORITY CODE PASSENGER          SERVICE
FOR EACH REQ IN PRT.UNSAT.PEQ WITH PAX.PRIORITY(REQ) GT 3
```

```
DO
    IF NLT.TIME(REQ) GT 36.0
        REMOVE REQ FROM PRT.UNSAT.REQ
        FILE REQ IN PENDING.REQUESTS
    ELSE
    IF NLT.DATE(REQ) GT YEAR.DAYS
        LET NLT.DATE(REQ)=NLT.DATE(REQ)-YEAR.DAYS
    ALWAYS
    IF NET.DATE(REQ) GT YEAR.DAYS
        LET NET.DATE(REQ)=NET.DATE(REQ)-YEAR.DAYS
    ALWAYS
    LET X=TRUNC.F(NET.TIME(REQ))
    LET Y=NET.TIME(REQ)-X
    LET NET.TIME(REQ)=(X+Y*6/10)*100
    IF NET.TIME(REQ) GT 2400
    LET NET.TIME(REQ)=NET.TIME(REQ)-(TRUNC.F(NET.TIME(REQ)/2400))*2400
        IF NET.TIME(REQ) EQ 0
            LET NET.TIME(REQ)=2400
        ALWAYS
    ALWAYS
    LET X=TRUNC.F(NLT.TIME(REQ))
    LET Y=NLT.TIME(REQ)-X
    LET NLT.TIME(REQ)=(X+Y*6/10)*100
    IF NLT.TIME(REQ) GT 2400
    LET NLT.TIME(REQ)=NLT.TIME(REQ)-(TRUNC.F(NLT.TIME(REQ)/2400))*2400
        IF NLT.TIME(REQ) EQ 0
            LET NLT.TIME(REQ)=2400
        ALWAYS
    ALWAYS
    WRITE
        R.ORIGIN(TRAVEL.REQUEST),
        R.DESTINATION(TRAVEL.REQUEST),
        NET.DATE(TRAVEL.REQUEST),
        NET.TIME(TRAVEL.REQUEST),
        NLT.DATE(TRAVEL.REQUEST),
        NLT.TIME(TRAVEL.REQUEST),
```

```
          PAX.LOAD(TRAVEL.REQUEST),
          PAX.PRIORITY(TRAVEL.REQUEST),
          DV.CODE(TRAVEL.REQUEST),
          PAX.1NAME(TRAVEL.REQUEST),
          PAX.2NAME(TRAVEL.REQUEST),
          PAX.RANK(TRAVEL.REQUEST)
          AS S 1,A 8,A 7,I 3,I 5,I 4,I 5,I 4,I 5,I 7,S 3,A 10,A 7,A 4,/
     ALWAYS
LOOP
ALWAYS
IF N.PENDING.REQUESTS GT 0
PRINT 5 LINES THUS

TRAVEL REQUESTS THAT MAY STILL BE SUPPORTED IN THE FUTURE

ORGIN   DEST    NET NET  NLT NLT  NO.   PAX      DV  NAME-FIRST         RANK/
ICAO-ID ICAO-ID DAY TIME DAY TIME PAX PRIORITY CODE PASSENGER          SERVICE
FOR EACH REQ IN PENDING.REQUESTS
DO
     IF NLT.DATE(REQ) GT YEAR.DAYS
          LET NLT.DATE(REQ)=NLT.DATE(REQ)-YEAR.DAYS
     ALWAYS
     IF NET.DATE(REQ) GT YEAR.DAYS
          LET NET.DATE(REQ)=NET.DATE(REQ)-YEAR.DAYS
     ALWAYS
     LET X=TRUNC.F(NET.TIME(REQ))
     LET Y=NET.TIME(REQ)-X
     LET NET.TIME(REQ)=(X+Y*6/10)*100
     IF NET.TIME(REQ) GT 2400
     LET NET.TIME(REQ)=NET.TIME(REQ)-(TRUNC.F(NET.TIME(REQ)/2400))*2400
          IF NET.TIME(REQ) EQ 0
               LET NET.TIME(REQ)=2400
          ALWAYS
     ALWAYS
     LET X=TRUNC.F(NLT.TIME(REQ))
     LET Y=NLT.TIME(REQ)-X
```

```
    LET NLT.TIME(REQ)=(X+Y*6/10)*100
    IF NLT.TIME(REQ) GT 2400
    LET NLT.TIME(REQ)=NLT.TIME(REQ)-(TRUNC.F(NLT.TIME(REQ)/2400))*2400
        IF NLT.TIME(REQ) EQ 0
            LET NLT.TIME(REQ)=2400
        ALWAYS
    ALWAYS
    WRITE
        R.ORIGIN(TRAVEL.REQUEST),
        R.DESTINATION(TRAVEL.REQUEST),
        NET.DATE(TRAVEL.REQUEST),
        NET.TIME(TRAVEL.REQUEST),
        NLT.DATE(TRAVEL.REQUEST),
        NLT.TIME(TRAVEL.REQUEST),
        PAX.LOAD(TRAVEL.REQUEST),
        PAX.PRIORITY(TRAVEL.REQUEST),
        DV.CODE(TRAVEL.REQUEST),
        PAX.1NAME(TRAVEL.REQUEST),
        PAX.2NAME(TRAVEL.REQUEST),
        PAX.RANK(TRAVEL.REQUEST)
        AS S 1,A 8,A 7,I 3,I 5,I 4,I 5,I 4,I 5,I 7,S 3,A 10,A 7,A 4,/
LOOP
ALWAYS
RETURN
END ''OF PRINT.SCHEDULE
```

VITAE

George P. Milne was born in Hot Springs, South Dakota on October 10, 1944. He graduated from high school in Edgemont, South Dakota in 1963. In 1967 he graduated from the United States Air Force Academy with a major in Aeronautical Engineering and a commission in the United States Air Force.

After completing navigator training in May 1968 he navigated the EC-121R in Southeast Asia and the C-130 at Forbes AFB, Kansas. He later instructed navigator students in the T-29 at Mather AFB, California. When the T-43 was introduced as the new navigation trainer, he was appointed as an initial flight examiner. In his last assignment he navigated JC-130s at Hickam AFB, Hawaii and was an operations officer assigned to the unit's Operations and Plans Division.

He received a Master's degree in Public Administration from Golden Gate University in 1974 and entered the Air Force Institute of Technology in August 1977.

He is married to Linda A. Bjerkelund, who was living in Colorado when they met. They have one son, Steven.

Permanent Address: Box 780
Edgemont, South Dakota 57735

Roger K. Coffey was born on 7 January 1945 in Washington, D.C. He graduated from high school in Roanoke, Virginia in 1963. He attended the United States Air Force Academy. Upon graduation in June 1967, he received the degree of Bachelor of Science and a commission in the USAF.

He completed pilot training in September 1968. Since that time, he has served as a C-141 pilot at Travis AFB, California; as an HH-43 pilot at Phan Rang AB, Republic of South Vietnam, and at Nellis AFB, Nevada; and as a C-5 flight examiner pilot at Travis AFB. He entered the School of Engineering, Air Force Institute of Technology, in August 1977.

He is married to the former Alice Rucker of Roanoke, Virginia. They have one son, Christopher.

Permanent address: c/o R. A. Coffey
5826 Santa Anita Terrace NW
Roanoke, Virginia 24012